AIエージェント
開発・導入ガイド

日本テラデータ株式会社

teradata.

目次

第1章：
AIエージェントとは

近年、データ量と処理速度の爆発的拡大を背景に、企業は膨大な情報の海から意思決定に必要な知見を瞬時に引き出す手段を切望してきました。従来のAIや機械学習（ML）は、大量データからパターンを学習し予測モデルを構築することで、需要予測や異常検知といった定型的なタスクを自動化してきました。しかし、これらはあくまで「統計的推論」に留まり、ユーザーの意図を理解して対話的に知見を提供するまでには至っていませんでした。

AIエージェントはこのギャップを埋める存在として登場しました。膨大なドキュメントの中から関連情報を検索し、必要に応じて外部システムを操作しながら、自律的にマルチステップの思考プロセスを実行できます。ユーザーは自然言語で問いかけるだけで、エージェントが内部でデータ抽出、分析、レポート生成、さらにはチケット起票やメール送信といったアクションまで一気通貫でこなすため、専門家でなくとも高度な業務自動化を享受できるようになります。

AIの能力進化を体系化するため、OpenAIは五つの段階を定義しています。レベル1「チャットボット（会話型AI）」では、質問に対して対話形式で回答を返すシステムが該当し、カスタマーサポートや社内FAQの自動化が実現されています。レベル2「推論」は人間レベルの問題解決能力を持つAIで、複雑なクエリに対して論理的思考を用いて段階的に解答を導き出します。

ここでは単純な応答を超え、背景情報や文脈を考慮した高度な判断が可能となります。

レベル3「エージェント」こそが本書で扱うAIエージェントのコアとなる段階です。このレベルでは、AIが自ら行動を起こせるシステムとして機能し、外部ツールの呼び出し、データベース照会、API連携、レポート生成、メール送信やチケット作成までを自律的に実行します。ユーザーは目標を伝えるだけで、必要なステップを分解し、実行し、結果を検証するプロセスをAI側が担います。

さらに進んだレベル4「イノベーター」では、発明を支援できるAIとして、既存の知見を組み合わせて新しいアイデアや解決策を提案する能力を備えます。研究開発や商品企画において、過去のデータだけでなく創造的な提案まで生み出せるフェーズです。最終段階のレベル5「組織マネージメント」は、組織全体の業務を遂行できるAIで、戦略立案から実行管理、リソース配分、パフォーマンス評価まで、経営判断に近い機能を担います。複数の部門や業務プロセスを横断して最適化を図り、組織運営そのものをサポートする未来像を示しています。

現在のAIエージェントは主にレベル3の段階にあり、自律的な行動実行による業務効率化と意思決定支援で大きな価値を発揮する可能性を秘めています。本書では、このレベル3のAIエージェントをいかに設計・構築し、企業のデータ活用推進に活用するかを詳細に解説します。

第2章：
AIエージェント vs エージェント型AI

「AIエージェント」と「エージェント型AI」これらの用語は似ているように聞こえるかもしれませんが、AIエージェントについて語る上で、実は大きく異なる概念であることを理解することが重要です。この2つの違いを正確に理解することは、AI活用戦略を検討する上で必要不可欠です。

	AIエージェント	エージェント型AI
定義	自然言語で指示を理解し、自律的にタスクを実行	複数のAIモジュールが協調して動作するシステム
特徴	・対話型インターフェース ・学習と適応能力 ・業務に特化した判断	・分散処理による高性能 ・複雑タスクの分割処理 ・システム間協調
利用例	顧客サポート、レポート生成、意思決定支援	製造プロセス制御、IoT統合管理、複雑な最適化

現在のAI活用の主流： AIエージェント

AIエージェントは、デジタル環境やリアル環境で状況を知覚し、意思決定を下し、アクションを起こすためにAI技術を適用した自律的または半自律的なソフトウェアです。具体的には、ユーザーの指示に基づいて特定のタスクを実行し、カスタマーサポートや業務自動化といった比較的限定された範囲での効率化を担います。

現在のAIエージェントの多くは、ある程度の判断力を持ち、シンプルなタスクの一部を自律的に実行できるものの、その機能は限定的です。たとえば、チャットボットが顧客からの問い合わせに対して事前に設定されたルールに基づいて応答する場合、これは典型的なAIエージェントの例と言えます。

AIエージェントは、一般に大規模言語モデル（LLM）を中核とするものの、その業務知識や専門的または、暗黙知を用いる推論には制限があります。従来のチャットボットがユーザーからの継続的な入力を必要とするのに対し、AIエージェントは利用可能なツール、メモリ、推論を備え、時間の経過とともにユーザーの期待に適応することを学習します。

次世代の自律的AI システム：エージェント型AI

一方、エージェント型AIは、組織のために行動し、自律的に意思決定を下してアクションを起こすために、組織に代わって行動する権利を付与された、目標主導型のAIです。このAIは、記憶、計画、知覚、ツール利用、そしてガードレールといったコンポーネントを組み合わせて活用し、複雑なタスクを完了し、最終的な目標を達成することを目指します。

エージェント型AIは、AIエージェントの「進化系」に位置づけられています。エージェント性と目標指向性を備え、記憶や計画、ツール活用などの高度な機能により、複雑なタスクを自律的に目的指向で遂行することが期待されています。

特に重要なのは、エージェント型AIが複数のAIエージェントが協調して動作するより高度なシステムであり、単一のエージェントでは対処できない複雑な問題を解決する能力を持つことです。人間からの直接的な指示なしに動作し、最終目標達成のために後続のアクションを事前に決定できる点が大きな特徴です。

本質的な違いは、自律性のレベルと協調性

この2つの概念の本質的な違いは、自律性のレベルと協調性にあります。AIエージェントは、人間の指示に基づいて動作し、「知覚→推論→行動→学習」のサイクルで動作しますが、エージェント型AIは、「計画」「記憶」「ツール利用」の能力が強化され、より複雑な問題解決や長期的な目標達成が可能です。他方、協調性と複雑性については、AIエージェントが比較的単純なタスクを個別に処理するのに対し、エージェント型AIは、複数のAIモジュールが協調して動作し、マルチエージェントシステムとして機能します。

実用的な使い分け

AIエージェントは、予測可能性と制御を必要とする反復的なルールベースのタスクに最適です。一方、エージェント型AIは、適応性、回復力、自律的な意思決定が求められる環境に適しています。

この違いを理解することで、企業は自社のニーズに応じて適切なAI技術を選択し、段階的にAIドリブン経営を実現することが可能になります。

次章では、具体的な開発手法を段階的に見ていきましょう。

第3章：
AIエージェント開発ステップ

AIエージェントの開発プロセスは、これまでのソフトウェア開発やレポート作成とは大きく異なります。まず従来の開発では要件定義を行い、画面設計、データベース設計、実装、テスト、リリースといったウォーターフォール型の段階を順番に踏むのが一般的でした。データ活用においても、BIツールでダッシュボードを作成し、定期的にレポートを生成するといった「定型化された作業」の自動化に留まっていました。

一方、AIエージェント開発では、対話インターフェースを通じた「目標達成プロセスの自動化」が主眼となります。最初に「ユーザーが何を達成したいのか」をシンプルな自然言語で定義し、その目標を実現するためのマルチステップフローを迅速にプロトタイピングします。ここでは画面設計ではなく、ノードという処理単位をつなぎ合わせることで、情報検索、データ抽出、分析、外部API呼び出し、成果物生成といった工程を柔軟に組み替えられる点が特徴です。

データ接続も従来と異なり、個別SQLを書くのではなく、コネクタ設定でさまざまなデータソースやドキュメントリポジトリを一元的にインデクシングできます。これにより、ユーザーの問いに即座に関連情報を検索し、LLMに渡すためのコンテキストを自動で整えられます。応答の質を高めるためのRAG（Retrieval-Augmented Generation）パイプラインは、一度構築すれば新しいドキュメントの追加も自動反映され、常に最新のナレッジベースを活用できます。

さらに、外部システムとの連携はコードを書くのではなく、あらかじめ用意されたAPI呼び出しノードや関数ノードを配置するだけで実現します。メール送信、チケット発行、BIダッシュボード更新など必要なアクションをノード間でつなぐだけで自動化できるため、従来のプログラミングに比べて導入速度が飛躍的に向上します。

開発サイクルも短く、小さなPoCから始めてユーザーフィードバックを反映しながら迅速にチューニングを重ねるアジャイルスタイルが基本です。これにより、初期段階から実務での価値を確認しつつ、段階的に機能を拡張していくことができ、従来の一括開発・一括リリース型のリスクを大幅に低減します。結果として、技術的な専門知識を持たないユーザーでも自らワークフローを設計・改善できる、データドリブン組織への移行を加速させる開発体験が実現します。

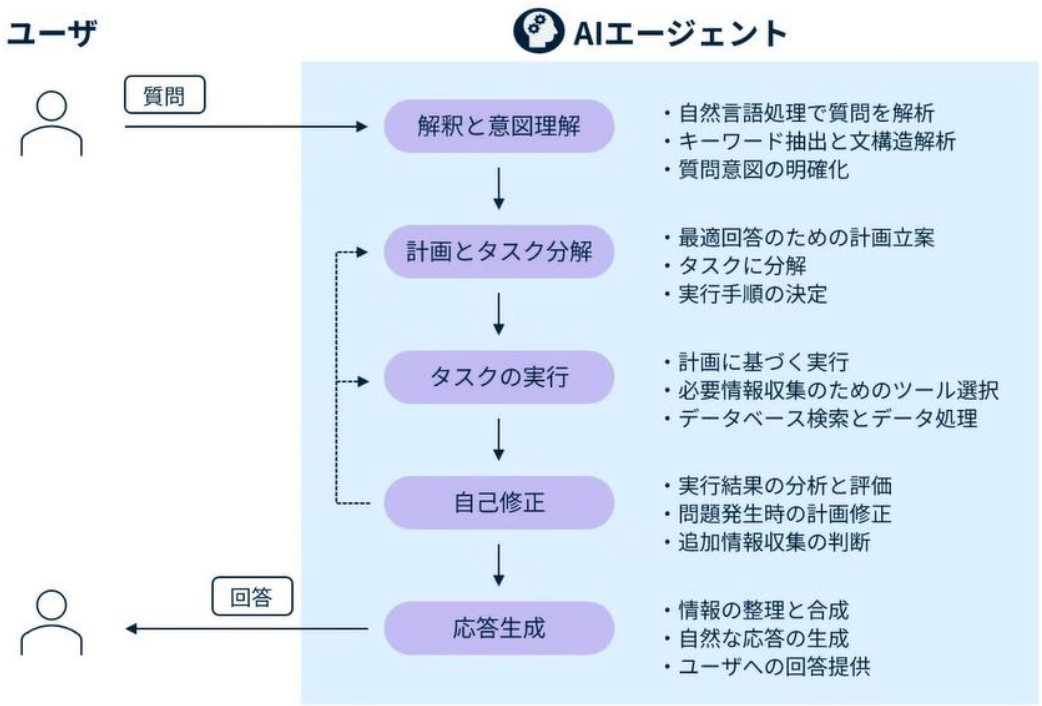
エンタープライズ向けエージェント型AI開発プラットフォームを選択し、開発することになりますが。代表的な開発ステップは下記となります。

Step1. ノードベースUIでのワークフロー設計

Step2. RAG連携

Step3. 外部機能連携とUIの生成

本章ではこれらについて解説していきます。



第3章： AIエージェント開発ステップ
Step1.ノードベースUIでのワークフロー設計

エージェント型AIアプリケーション開発の出発点は、業務の流れを“ノード”という最小単位に分解し、それらをつなぎ合わせるAIエージェントワークフローの設計です。ノードベースUIは、この作業を視覚的かつ直感的に行えるため、コードを書かず、更に、他のツールでは数個から10個程度ノードが必要となる複雑な処理も1つのノードで組み立てられることが最大の特徴です。

まず、ユーザーが自然言語でエージェントに期待するゴールを明確にします。「今週優先すべき営業案件を一覧化し、リスク要因を分析してアクション案を示してほしい」といった具体的な要求を起点に、必要となる処理ステップを洗い出します。たとえば、このケースでは

1. **ユーザーの問いを受け取る入力ノード**
2. **商談データベースから関連レコードを抽出するクエリノード**
3. **データを分析して優先順位とリスク要因を算出する分析ノード**
4. **レポートを生成する生成ノード**
5. **結果をメールで配信するアクションノード**

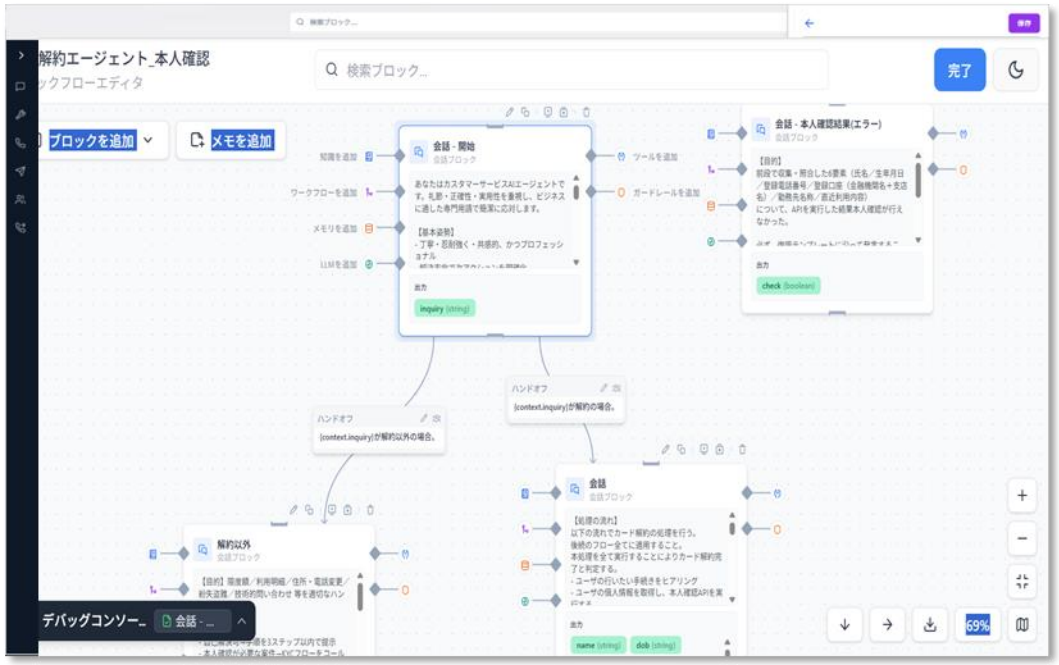
このように、各処理を担当するノードに機能を割り当てます。

ワークスペース上では、これらのノードがドラッグ＆ドロップで配置でき、矢印や線でつなぐだけでデータの受け渡しが定義されます。ノードひとつひとつには名称と役割を付与し、インプットとアウトプットのデータ形式を設定します。たとえば、抽出ノードの出力スキーマとして「案件ID、商談金額、ステージ、最終更新日時」というフィールドを定義しておけば、その後段の分析ノードは必要なフィールドを自動で参照し、指定したアルゴリズムを適用できます。こうしたスキーマ設計により、ノード間の情報齟齬を防ぎ、ワークフロー全体の堅牢性を高められます。

さらに、ノードベースUIは各処理ロジックを即座にテストできるインタラクティブデバッグ機能を備えています。たとえば、抽出ノードに検索条件を入力して「テスト実行」ボタンを押せば、実際のデータサンプルが返ってきてスキーマが正しいかを確認できます。分析ノードや生成ノードについても、モック入力を与えれば期待する分析結果やレポートが得られるかをすぐに検証できるため、開発初期の段階から品質を担保した設計が可能です。

また、分岐やループなど条件付きのフローも可視化できます。営業案件のうちリスクが高いものだけを別ルートに回し、特定のマネージャーへ個別通知する処理を追加したい場合、そのルールを判定ノードとして挿入し、条件を満たす場合は通知ノードへ、満たさない場合は通常のレポート生成ノードへと線を分岐させるだけで実装できます。このように、ビジュアルキャンバス上でワークフローを“見ながら作る”ことで、複雑な業務ロジックを誰もが理解しやすく、またレビューや改善を迅速に行える環境が実現します。

最後に、完成したワークフローはドキュメント化せずともプラットフォーム内で自動的にバージョン管理されます。誰がいつどのノードを変更したかが履歴として残るため、開発チーム内での共同作業や過去バージョンへのロールバックもスムーズです。こうしたノードベースUIによるワークフロー設計は、AIエージェント構築のスピードと信頼性を飛躍的に向上させる基盤となります。



第3章： AIエージェント開発ステップ
Step2.RAG連携

「ただのAIチャット」ではなく、企業固有の知識や最新データを反映した高品質な回答を実現するために不可欠なのがRetrieval-Augmented Generation（RAG）です。RAGは、大量のドキュメントやデータベースを検索し、その結果を生成モデルに組み込むことで、根拠ある回答と説得力のあるレポートを同時に提供できます。

まず最初に、接続したいデータソースをプラットフォームに登録します。これにはDWH、ERPやCRMのデータベース、社内ファイルサーバーに保管されたPDFやPPT、さらには外部のマニュアルやWebコンテンツまでが含まれます。これらの資料を自動的に走査し、タイトルや作成日、所有部門といったメタデータを付与しながらインデクシング処理を行うことで、検索時のフィルタリング精度を高めます。

次に、登録されたドキュメントを埋め込みモデルでベクトル化し、ベクトルストアに格納します。この段階で文書を小さな断片に分割し、それぞれを高次元のベクトルに変換。ベクトルストアは高速な近傍探索に最適化されており、ユーザーからのクエリを同様にベクトル化したうえで「最も近い」文書断片を瞬時に抽出します。

ユーザーが自然言語で質問を投げかけると、システムはまずその質問を高度なコンテキスト分析を行い、必要であれば、逆質問を生成し、適格な対応を進め、ベクタークエリを発行し、ベクトルストアに照会します。ユースケースにより、グラフナレッジも併用することで得られた関連度の高い文書断片は、回答生成のためのコンテキストとして用いられます。例えば「先月の売上トレンドと要因分析をレポート化してください」と尋ねられた場合、プラットフォームは売上データのダッシュボードや関連部署の分析レポートなどから該当する情報を抜き出し、その断片を一連のシステムプロンプトに組み込みます。

プロンプトには、抜き出した文書断片だけでなく、Few-shot事例やガイドラインを含めることで、生成モデルに一貫した出力スタイルと精度を担保します。こうして構築されたコンテキストプロンプトをもとに、モデルは文章を生成し、必要に応じてグラフや表の作成指示も行って最終的なレポートを組み立てます。さらに、生成結果には必ず参照元のドキュメントページや節番号を付加し、ユーザーが情報の正確さをすぐに検証できる仕組みを保持します。

運用面では、頻出クエリや計算リソースの多い生成タスクをキャッシュ（メモリー、あるいはDB）に保存し、同じ要求に対しては即座に応答できるようにします。また、新しいドキュメントが追加された際には差分インデクシングを自動化し、プラットフォーム上のベクトルストアにリアルタイムで反映。これにより、常に最新のナレッジをRAGパイプラインに組み込んで、回答の信頼性を維持できます。

このようにRAG連携を通じて、単なる言語モデルでは実現不可能な「企業固有のナレッジ統合」と「アクションにつながるレポート生成」を両立させます。次のステップでは、こうして得られた知見を実際の業務アクションに結びつける外部ツール連携について詳しく見ていきましょう。

第3章： AIエージェント開発ステップ
Step3.外部機能連携

エージェント型AIの真価は、データの検索と生成だけに留まらず、ビジネスプロセス全体を自動的に動かせる点にあります。そのためには既存システムや外部サービスとのシームレスな連携が不可欠です。

まず、AIエージェントはAPI呼び出しを通じてERPやCRM、社内データベースなどの業務システムからリアルタイムに情報を取得したり、更新したりできなければなりません。その実現手段として、プラットフォーム上に用意された外部ツール登録機能やワークフローを活用します。エンドポイントごとにHTTPメソッドや認証トークンを設定し、パラメータテンプレートを定義しておけば、ユーザーが自然言語で指示を出すだけでAIエージェントが内部的にAPIリクエストを発行し、必要なデータを引き出します。

次に、複雑なデータ処理や集計が必要なケースでは、カスタム関数ノードを利用します。ここにはSQLクエリやPythonスクリプト、あるいは社内向けのマイクロサービスを紐付けておけるため、汎用的なノードだけで対応できない高度な計算やデータ変換も、多様な言語で記述して取り込むことが可能です。たとえば、CRMの商談データを特定のアルゴリズムでスコアリングし、その結果を別の分析モジュールに渡して判断材料とする一連の流れをノードチェーンで可視化できます。

さらに、生成されたアウトプットを関係者に確実に届けるために、メール送信やチャット通知、チケット作成といった外部アクションを設定します。メールノードであれば送信先アドレスや件名、本文テンプレートをドラッグ＆ドロップで配置し、生成結果を自動的に埋め込むように定義。SlackやMicrosoft Teamsへの通知ノードを挿入すれば、分析結果や重要アラートをリアルタイムに現場へ共有できます。また、JiraやServiceNowなどのチケットシステムと連携し、異常検知レポートを自動で課題管理ツールに登録することで、対応漏れを防ぎつつ迅速な問題解決を後押しします。

こうした外部機能連携において重要なのは、エラー発生時のリカバリ設計です。API呼び出しがタイムアウトした場合や外部サービスが応答しない場合には、あらかじめ定義したリトライポリシーが働き、必要に応じ代替ノードへ処理を引き継ぎ、フェールセーフを実現します。ユーザーへの通知やログへの書き込みを組み込むことで、障害発生時にも迅速に事態を把握して復旧措置を講じることができます。

また、セキュリティとガバナンスの観点では、APIキーやシークレット情報はVault（専用のシークレット管理システム）に格納し、ノードからは参照のみを許可します。実行権限はロールベースのアクセス制御（RBAC）で厳格に管理し、誰がどの外部機能を呼び出せるかを細かく設定します。これにより、機密データの漏洩リスクを最小化しつつ、信頼性の高い自動化フローを運用できるのです。

更に、UIの開発も同じプラットフォームで行うため、自然言語によるUIのコード生成とAIエージェント・リストからアサインを自動で行い、エージェント型AIとして稼働するUIを生成・動作確認することができます。

外部機能連携と使いやすいUIを通じて、単なる情報検索ツールから脱却し、実際の業務アクションを自動で実行する真のビジネスオートメーション基盤へ昇華します。

第3章： AIエージェント開発ステップ
補足.ガードレイル機能

エージェント型AIが業務フローを自律的に実行するにあたり、誤った情報の入手や不適切な対話・回答を未然に防ぐ「ガードレイル機能」は欠かせません。ガードレイルは道路のガードレールのように、想定外の逸脱を防止し、安全かつ信頼性の高いサービスを支える仕組みです。

まずガードレイルの主な役割は、禁止トピックの制御、出力制限、意図確認・再確認、ツール制御の四つに集約されます。禁止トピックの制御では、特定のキーワードやセンシティブな内容を回避させます。出力制限では、長さやトーン、形式（例：HTML禁止など）を指定して、出力の正誤性を保ちます。意図確認・再確認では、ユーザーのリクエストが曖昧な場合にユーザーのリクエストが曖昧な場合に「この質問は〇〇について話していますか？」と確認を行います。そして、これらのポリシーはセッションをまたいで維持されます。ツール制御では、顧客ID・認証情報など、必要な入力がそろっているときのみ、DB等の外部システムへの連携を許可します。

これらを実装するには、エージェントのノードに付帯するガードレイルを開きます。単なるプロンプト制御だけでなく、ガードレイルを複数レイヤー設定することも、またさらに詳細な制御のため、Pythonのプログラムを埋め込むことも可能です。出力前/出力後/ツール制御/セッション維持/ログ記録といった複数層で構成された技術的セーフティネットにより、安全性と規制遵守が担保されます。

さらに、開発段階ではガードレイルをテスト用ワークフローに適用し、意図的に不正な入力や不正なツールアクセスをシミュレーションしてポリシーが正しく機能するかを検証します。テストが完了したら、本番ワークフローにポリシーをバインドし、監査ログの保存先を指定。これにより、誰がいつどのガードレイルに抵触したかが追跡可能になります。

最後に、ガードレイル設定は運用中にも継続的に見直しが必要です。新たな脅威や業務変更に応じてアクセス権やバリデーションルールを更新し、定期的にポリシー適用のテストを実施します。こうしたガードレイル機能の適切な設定と運用により、AIエージェントは安全かつ信頼性の高い自動化基盤として組織に定着し、ビジネス成果を確実に支えます。

第4章：
AIエージェント時代のデータ基盤要件

エージェント型AIが社内の主要業務を自律的に実行し始めると、従来のBIやレポート基盤では想定していなかったクエリ増加と問い合わせパターンの複雑化が顕在化します。たとえば、ある営業チームが毎朝「優先案件の最新状況とリスク評価を一覧化して」と複数のエージェントに依頼すれば、通常の月次レポート数十件分を超えるAPIコールとデータベースクエリが瞬時に発生することになります。これらの急激な負荷増加に耐えられないシステムでは、レスポンス遅延やタイムアウトが頻発し、ユーザー体験が著しく損なわれるだけでなく、業務への信頼も失われかねません。

下記のような課題が顕在化することが予測され、あらかじめ対応可能なデータ基盤を選択することが重要となります。

1. クエリ量の急増と複雑化

ユーザーが日々の業務をエージェントに任せ始めると、定期レポートの数十倍に相当するクエリが同時並行で発生します。しかも単純なSQL実行ではなく、ベクトル検索や生成モデルへのコンテキスト作成を含む多段階処理が求められ、従来のストレージやインデックス方式ではレスポンス遅延やタイムアウトが頻発してしまいます。

2. 多様なデータソース統合とリアルタイム更新

AIエージェントはERP／CRMデータ、ドキュメントファイル、ログ、APIなどを横断的に検索し活用するため、データソースごとに個別設計したパイプラインではメンテナンス負荷が高くなります。また、ナレッジの鮮度を保つための差分インデクシングやリアルタイム更新にも対応できる設計が必須です。

3. 高速ベクトル検索とスケーラビリティ

数百万〜数千万件規模の埋め込みベクトルをミリ秒単位で検索する能力が不可欠です。単一ノードではなく分散環境でのベクトルストア、GPUやASICアクセラレーションを活用したインデックス構造の最適化が求められます。

4. マルチモデル推論とコスト管理

テキスト生成には大規模言語モデル、分析にはカスタム予測モデル、集計には時系列エンジン…と用途ごとに最適なモデルを切り替えられる推論プラットフォームが必要です。動的なモデルロードバランシングやバージョン管理、利用状況に応じたコスト最適化機能も重要です。

5.キャッシュとプリフェッチによるパフォーマンス向上

頻出クエリを自動検知してホットデータをキャッシュし、さらにユーザー行動を予測して次回クエリを事前計算・キャッシュする機能により、繰り返し発生する検索リクエストのレスポンスを飛躍的に高速化します。

6. APIゲートウェイとオートスケール

外部システムとの連携やユーザーからの問い合わせをAPI層で一元管理し、認可認証、トラフィック制御、DDoS対策を実装。バックエンドのオートスケール機能により、ピーク時にも安定した処理能力を維持できることが重要です。

7. 包括的モニタリングとアラート

クエリ数、レイテンシ、エラー率、リソース使用状況などをリアルタイムに可視化し、しきい値超過時には自動アラートを発出。ログを収集・解析することで、ボトルネックや異常動作の原因を即時に特定し対処できる体制を整えます。

8. 厳格なセキュリティとガバナンス

機密データを扱うシステムでは、アクセス制御（RBAC）、通信暗号化、シークレット管理、監査ログの長期保存が必須です。特に、誰がいつどのデータにアクセスし、どのアクションを実行したかを可視化できる仕組みがリスク低減につながります。

9. 継続的キャパシティプランニング

本番運用での負荷増大やサービス拡張を見越し、定期的な負荷試験とキャパシティ評価を繰り返し実施。インフラ構成やリソース配分を機動的に見直して、常に余裕をもった運用を維持します。

これらをすべて満たすデータ基盤は、自前構築では膨大な工数と専門知識を要し、運用コストや技術的負債が増大するリスクがあります。一方で、これらの要件をネイティブに備え、導入から運用までを一気通貫で提供するソリューションを採用することで、AIエージェントを本番環境で安定的に運用し、急増する複雑なクエリにも対応しながら、ビジネス価値を最大化できるのです。

第5章：
AIエージェントの導入ステップ

企業内にエージェント型AIアプリケーションを本格導入するには、戦略立案から運用定着までの一連のステップを計画的かつ実践的に進めることが成功の鍵となります。ここでは、DX部門やIT部門のマネジメント層が押さえるべき主要工程を、豊富な実例とともに詳述します。

1. ビジネスゴールの明確化

まず最初に行うべきは、ビジネスゴールの明確化です。現場の課題をヒアリングし、「何を自動化し、どのKPIで価値を測るか」を定量的に設定します。たとえば「営業案件の優先度精度を現状比20%向上」「問い合わせ対応時間を50%短縮」といった具体的指標が必要です。ゴール設定後は、ステークホルダーを横断的に巻き込み、プロジェクトのロードマップとガバナンス体制を確立します。

2. PoCの実施

次に、PoC（概念実証）フェーズに移行します。ここでは小規模かつ短期間で価値が見込めるユースケースを選定し、要件定義からワークフロー設計、RAG連携、外部連携までを含む小規模プロトタイプを構築します。PoC実施中は、定期的にユーザーテストを行い、応答精度や処理時間、UXに関するフィードバックを集めて迅速に改善サイクルを回します。この段階で得られた成果と課題を、次の拡張フェーズの土台にします。

3. パイロットプロジェクトの実施

PoCで有効性が確認できたら、パイロット展開に進みます。ここでは対象業務範囲を部門単位に拡大し、利用ユーザーを増やして実運用に近い環境でテストします。運用に必要なログ収集、モニタリング、アラート設定を整備し、運用オーナーを定めてエスカレーションルールやリカバリシナリオを策定します。また、ガードレイル機能を本番ポリシーに切り替え、セキュリティとコンプライアンス要件を満たした運用を担保します。

4. 本番導入

パイロット展開の成功を受けて、本番スケールアウトフェーズでは全社横断的な導入を図ります。APIゲートウェイやオートスケーリング設定を最適化し、キャパシティプランニングに基づくインフラ拡張を実施。エージェントごとにSLA（サービスレベル指標）を定義し、継続的なモニタリングダッシュボードと運用レポートを公開して各部門へ透明性を提供します。

5. 継続的改善

さらに、継続的改善と組織文化への定着です。LLMOpsサイクルを通じ、プロンプトやワークフローのパフォーマンスデータを分析し、定期的に精度改善を実行します。さらに、エージェント利用に関するユーザー教育プログラムを展開し、成功事例を社内で共有することで、データドリブン文化を醸成します。

このように、導入から運用、改善、普及までを一気通貫で推進することで、AIエージェントは単なる技術実験に留まらず、企業の競争力を支える重要な基盤へと進化します。

第6章：
自然言語検索エージェントの可能性

すでにデータ基盤を導入し、BIツールやダッシュボードで可視化を実現している企業にとって、次なる挑戦は「蓄積したデータをいかに現場で活用し続けるか」です。従来型のデータ分析では、ユーザーは前提となる画面操作やクエリ言語の習熟が必要であり、専門チームへの依存度が高いままです。しかし自然言語検索エージェントは、このギャップを劇的に埋めます。

エージェント型AIが対話を実現

自然言語検索を実現するエージェント型AIは、日々の業務で最も身近なインターフェースである「対話」を通じてデータ分析を可能にします。ユーザーはSQLやBIツールの操作を意識することなく、「今期の売上推移と前年同期比較をグラフ化して」「地域別の離脱顧客数を要因ごとに要約して」「在庫回転率が落ちている製品とその背景を教えて」といった自然な問いかけを行うだけで、エージェントが裏側の複雑なクエリ生成と集計を瞬時に実行します。これにより分析リードタイムは数時間から数秒へと短縮し、現場担当者が自らデータにアクセスして意思決定を下す権限が与えられます。

しかも自然言語検索エージェントは単なる「検索」機能に留まらず、取得したデータの要約や傾向分析、さらには異常値検知や将来予測モデルまで組み込むことで、問いに対する「深みのある回答」を自動生成します。売上データをグラフ化するだけでなく、急激に増減した要因や取るべきアクション案まで提案するため、報告書作成や会議準備に費やす時間を大幅に削減し、その分を戦略立案や顧客対応といった価値創出に充てることができます。

自然言語検索の活用例

既存のデータ基盤に対して自然言語検索エージェントを追加導入すると、蓄積されたデータ資産は眠ることなく現場に流通し続けます。営業部門は成果指標をリアルタイムに問うことで顧客接点での迅速な意思決定を可能にし、マーケティング部門はキャンペーン効果やチャネルごとのROIを瞬時に把握して予算配分を最適化できます。製造・物流部門では、稼働率や品質トレンドを自然言語でモニタリングし、異常兆候をいち早く察知して稼働停止リスクを低減できます。こうした現場主導のデータ活用は、全社的なデータドリブン文化をより強固にし、組織のアジリティを飛躍的に高めます。

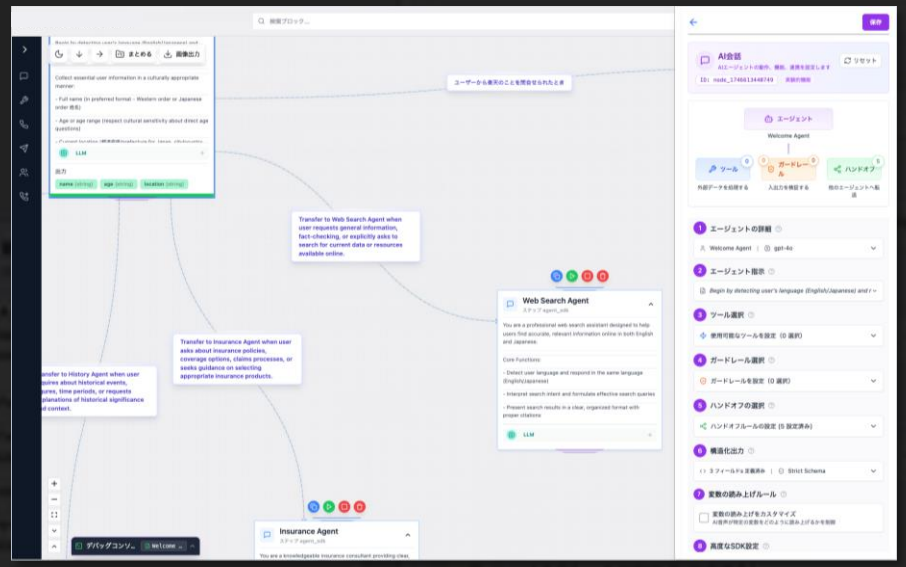
自然言語検索の利点

さらに、自然言語検索エージェントはユーザーごとの権限や役割に応じた最適化も得意です。管理職には全社KPIのダッシュボード要約を示し、担当者には自部門の詳細分析レポートを提供するといった具合に、同じ質問でも出力レベルを自動調整します。このパーソナライズ機能により、利用者は自分の業務に最も必要な情報だけを得られ、余計なデータに惑わされずに本質的な意思決定に集中できるのです。

導入ハードルも低い点が自然言語検索エージェントの魅力です。MCPサーバを用いて、既存のデータウェアハウスやデータレイクに接続するだけで、すぐに対話型インターフェースが使えるようになります。初期PoCから本番運用までの期間を従来の数カ月から数週間へと短縮し、その間に得られた現場のフィードバックを迅速に反映することで、エージェントが現場にフィットしたサービスへと進化します。

自然言語検索エージェントの導入は、単にデータ分析を手軽にするだけのツール追加ではなく、全社的なデータ活用の形を根本から変えるトリガーとなります。高度なデータ基盤を既にお持ちの企業ほど、その真価を最大限に引き出せるため、エージェントを「次の成長を支える中核技術」として位置付けることが強く推奨されます。

第7章：
AIエージェントプラットフォーム「AP-AI」



本書でご紹介した開発画面は、日本のAI開発企業AtPeak株式会社が提供するAIエージェントプラットフォーム「AP-AI」のものです。

AP-AIは、ノードベースUIやワークフロー設計、RAGパイプライン、プロンプト最適化、外部システム連携、UI生成、LLMOpsなど、エージェント型AIアプリケーションの開発と運用に必要な機能を一つのプラットフォームに統合しています。初期導入から本番運用、さらには継続的な改善フェーズに至るまで、散在するツールやサードパーティ製プラグインを組み合わせずに済むため、導入工数と学習コストを大幅に削減できます。

特徴1. 開発スピードの高速化

開発スピード面では、業務別のテンプレートとドメイン特化モジュールを用意しており、一般的な業務要件をコードレスで反映可能です。これに対し、通常のプラットフォームではテンプレートが汎用的に留まり、カスタマイズにはスクリプト作成や外部SDKの開発が必要になることが多く、初期PoCの立ち上げに要する期間が長期化しがちです。

特徴2. セキュリティとガバナンス

セキュリティとガバナンスはAP-AIの最重要要素として組み込まれています。組織ポリシーエンジンにより、ノード単位でデータアクセス権限を細かく制御できるほか、すべてのワークフロー実行とシークレット操作は自動的に監査ログに記録されます。これにより、外部のシークレット管理ツールやログ収集エージェントを別途統合する必要がなく、要件定義からコンプライアンス適合までを一気通貫で実現します。

特徴3. ハイブリッドな推論エンジン

推論エンジンはハイブリッド構成を採用し、オンプレミスGPUクラスタとクラウドベースの大規模言語モデルをシームレスに切り替えられます。この機能により、機密データは社内インフラ上で安全に処理しつつ、コストパフォーマンスを最適化できるため、常にコストとセキュリティのバランスを維持可能です。対して多くの開発プラットフォームでは、外部LLM利用時にすべてのデータがクラウドに送信されるため、機密情報の扱いに制約が生じるケースがあります。

特徴4. プロンプトの最適化

プロンプト最適化では、自動化された多変量テスト機能が利用でき、複数のシステムプロンプトやFew-shot例を同時に検証して最適解を自動で選定します。手動でのA/Bテストやスクリプトによるパラメータ調整を必要とせず、リリース直後から高い応答精度を担保できる点が大きな強みです。

特徴5. ノーコード外部連携

さらに、主要ビジネスアプリケーションやクラウドサービスと連携するためにMCPサーバに対応しており、MCPサーバのないシステムやサービスを除き、外部システム連携ノードの開発を一切不要とします。これにより、社内のERP、CRM、チケット管理、BIツールなどへの接続は数クリックで完了し、連携エラーやバージョン齟齬といった運用リスクも大幅に軽減されます。

特徴6. スケーラビリティ

最後に、サービスはマイクロサービスアーキテクチャで構築され、各コンポーネントが独立してオートスケール。大規模な本番環境でも高可用性を維持しつつ、利用状況に応じたリソース配分が自動で行われるため、運用負荷を最小限に抑えつつ継続的な拡張が可能です。

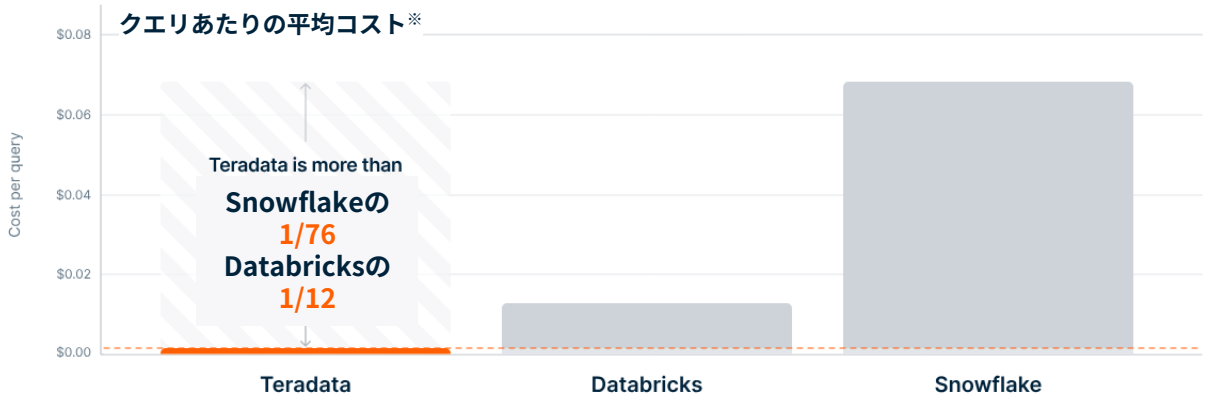
これらの特長により、AP-AIは短期間で高いROIを実現し、DX推進を加速させたい組織にとって最適な選択肢の一つと言えるでしょう。

第8章：
AIエージェント時代のデータ基盤「Teradata Vantage」

第5章で解説したように、AIエージェントの活用が進む中で、様々な課題が顕在化し、対応が必要となっています。

第5章では、1. クエリ量の急増と複雑化、2. 多様なデータソース統合とリアルタイム更新、3. 高速ベクトル検索とスケーラビリティ、4. マルチモデル推論とコスト管理、5. キャッシュとプリフェッチによるパフォーマンス向上、6. APIゲートウェイとオートスケール、7. 包括的モニタリングとアラート、8. 厳格なセキュリティとガバナンスの必要性について解説しました。

これらに対応し、現実的に運用し続けられる選択肢として、「大量クエリを低コストで処理できる」ことがプラットフォーム選定の最重要要件となります。



求められる要件に応える、ハイブリッドクラウドデータ基盤「Teradata Vantage」

Teradata Vantageは、求められる機能要件に対応するだけでなく、他の主要クラウドデータプラットフォーム（Databricks、Snowflake）と比較して圧倒的なコスト効率を実現します。この理由は、超並列処理（MPP）アーキテクチャと高度なコスト最適化機能にあります。

クエリ実行性能

Teradataは、Databricks比で8倍、Snowflake比で62倍の高速化を実証。同じリソース量でより多くのトランザクションを完了できるため、必要リソースを抑えつつ処理能力を最大化します。

クエリあたりの平均コスト

クエリ単位で見ると、Databricksの約1/12、Snowflakeの約1/76という低コストで実行できることが確認されています。仮にSnowflakeで76万円かかる処理を、Teradataでは1万円未満の投資で実行可能というインパクトです。

自動ワークロード最適化

インテリジェントチューニングがバックグラウンドで動作し、SQLの再コンパイルや統計情報の自動更新を行うことで、無駄なリソース消費を排除。運用管理者の手動チューニング負荷も大幅に軽減します。

リソース分離と優先度制御

ワークロードごとに優先度設定や隔離が可能。ビジネスミッションクリティカルな処理を確実にリソース確保しながら、アドホック分析は低コストモードで賄うことで、全体コストを最適化します。

ハイブリッドクラウド対応によるコスト最適化

オンプレミスとクラウドを透過的に統合できるため、定常的なトランザクションはオンプレミスの低TCO環境で処理し、一時的ピークやAI推論はクラウドの弾力性でまかなうなど、最適な費用配分が可能になります。

※このデータは、同等の構成システムにおける実際の混合分析ワークロードに基づいています。Teradataのワークロード比較の手法とプロセスの詳細については、www.teradata.jp/competitive-workload-comparisons をご確認ください。

第9章：
Teradataのご紹介

Teradata Corporationは、1979年に米国カリフォルニア州で誕生し、エンタープライズ向けハイブリッドクラウドデータプラットフォーム「Teradata Vantage」を中核に、世界中の大手企業にデータおよびアナリティクスのソリューションを提供してきました。

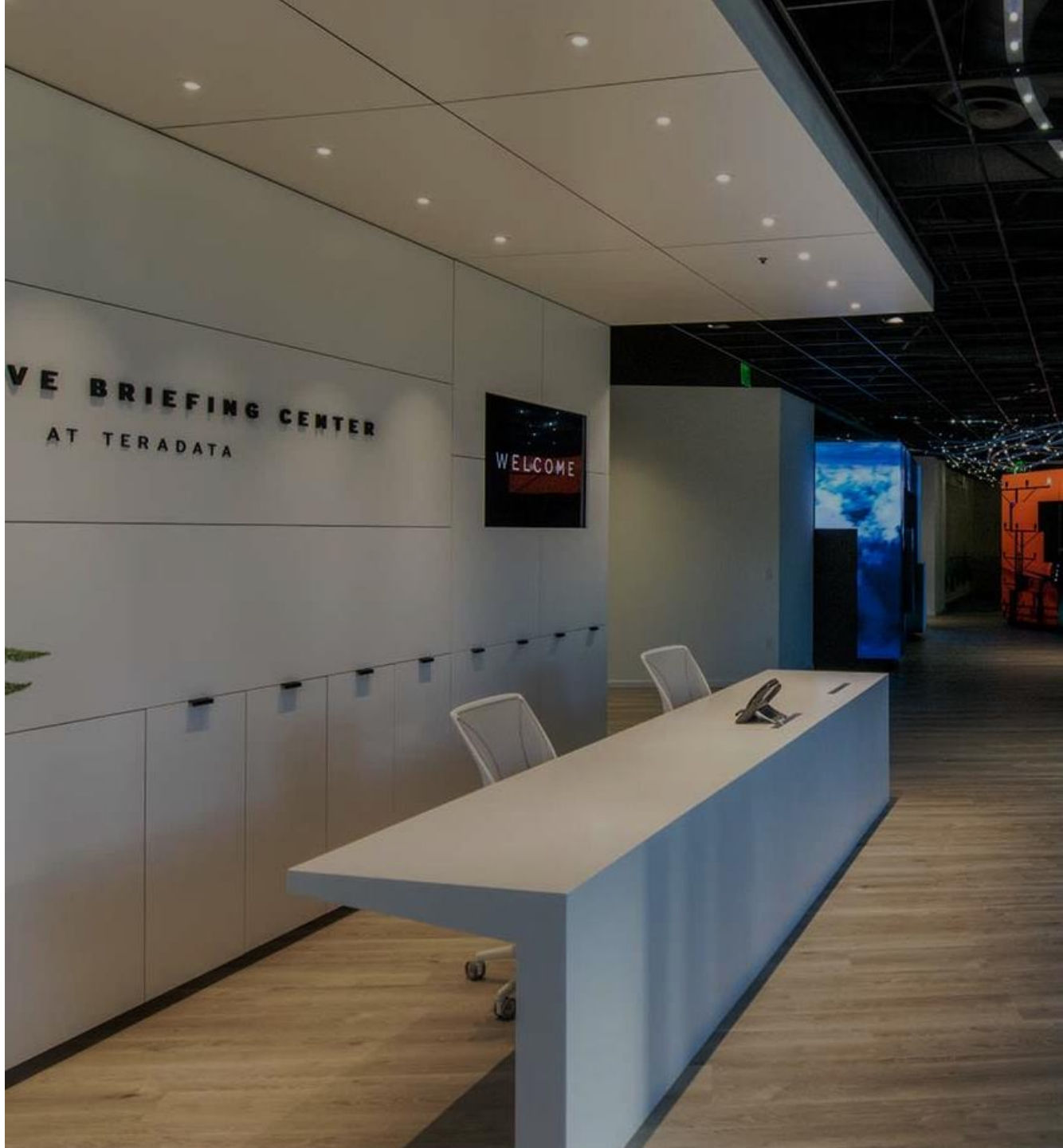
これまでの実績と最新技術の融合により、Teradataは、企業がデータとAIをフル活用して真のAIドリブン経営を実現するための最良のパートナーとなります。挑戦的なビジネス課題こそ、データが示す新たな未来への入口です。ぜひ次の一歩を、Teradataとともに踏み出してください。

Teradata Corporation 会社概要

本　　社	米国カリフォルニア州サンディエゴ
設　　立	1979年
代　　表	社長兼CEO Steve McMillan
従業員数	約6,500名
事業展開	41ヵ国
パートナー	100社以上

日本テラデータ株式会社 会社概要

本　　社	東京都港区赤坂2-23-1 アークヒルズ フロントタワー
代　　表	代表取締役社長 大澤 毅
設　　立	2007年（平成19年）4月20日
資 本 金	4億9千万円



teradata.