🤖 Создание AI-агентов с Claude в Microsoft Foundry

🧁 Построение AI-агентов с Claude в Microsoft Foundry

Ключевые тезисы:

- 🎯 Тренд смещается от простых диалогов с ИИ к созданию **агентных систем**, способных планировать, рассуждать и действовать.
- ⚙️ **Microsoft Foundry** — это единая платформа для создания AI-приложений и агентов в масштабе.
- 🤝 Интеграция **Claude** в Foundry даёт разработчикам лучшие в классе модели для рассуждений и готовые корпоративные функции.
- 🛠️ **Model Context Protocol (MCP)** — открытый стандарт для подключения агентов к внешним системам и инструментам.

📈 Эволюция AI: от чатов к агентам

Индустрия уходит от одношаговых AI-диалогов к созданию **агентных систем**. Такие системы могут:

- Выполнять **многошаговые рассуждения** в длинном контексте.
- Быть **надёжными, наблюдаемыми и безопасными** (критично для корпоративного использования).
- **Подключаться к различным инструментам и внешним системам** (базам данных, API и т.д.).

Чтобы реализовать этот потенциал, недостаточно просто улучшать модели — нужны **платформы для их исполнения**.

🏢 Что такое Microsoft Foundry?

Microsoft Foundry — это единая платформа для создания AI-приложений и агентов в масштабе.

Ключевые компоненты Foundry:
-## **Модели** (включая Claude).
-## **Сервис агентов** для их оркестрации.
-## **Инструменты и интеграции** (поддержка более 1400 встроенных коннекторов и MCP-инструментов).
-## **ML-сервисы** (например, тонкая настройка моделей).

Преимущества для предприятий:

- 🔒 Встроенная безопасность, наблюдаемость и управление.
- 🔗 Интеграция с Microsoft Defender, Purview и Entra ID.
- 🚀 Ускоренный цикл от разработки до продакшена благодаря встроенным инструментам оценки и мониторинга.

💡 Почему использовать Claude в Foundry?

Сочетание Claude и Foundry даёт разработчикам четыре ключевых преимущества:

1. ✅ **Лучшие в классе модели для рассуждений** (например, Claude Opus), отлично справляющиеся с планированием и длинным контекстом.
2. ✅ **Готовые инструменты для создания агентов** прямо на платформе.
3. ✅ **Корпоративные функции «из коробки»** (безопасность, соответствие требованиям).
4. ✅ **Быстрый путь в продакшен** благодаря предварительно собранной платформе.

🧪 Практический воркшоп: Агент для кондитерской "Sparkles"

Цель: построить AI-агента, который поможет кондитерской справляться с потоком заказов на кексы, используя Claude в Microsoft Foundry.

🚀 Шаг 1: Начало работы в Foundry

1. **Доступ к Foundry**: Войдите в платформу Microsoft Foundry.
2. **Работа с моделями**: В разделе `Build` > `Models` выберите модель (например, Claude Sonnet 4.6).
3. **Тестирование в Playground**: Используйте игровую площадку Foundry для экспериментов с моделью и системными промптами (например, «Вы — разумный кекс»).

🔧 Шаг 2: Подключение модели к локальной среде разработки

1. **Получение данных для API**: В Foundry на вкладке `Details` скопируйте **Target URI** и **API Key**.
2. **Настройка окружения**: В VS Code откройте файл `.env` и обновите переменные:
   - `ENDPOINT` (URI должен заканчиваться на `/anthropic`, без `/v1/messages`).
   - `API_KEY` (вставьте скопированный ключ).
   - `MODEL` (например, `claude-4.6-sonnet`).

🤖 Шаг 3: Создание агента с Microsoft Agent Framework

Microsoft Agent Framework — это открытый фреймворк для создания агентов (доступен на Python и TypeScript).

Базовый код агента:

```
# Импорт и инициализация клиента, который использует переменные из .env
from agents import Agent

# Определение агента
agent = Agent(
    name="cupcake_agent",
```

```
    instructions="You are a helpful cupcake ordering assistant."
)
```

После запуска агент готов к базовому взаимодействию в терминале.

🔌 Шаг 4: Подключение инструментов через MCP (Model Context Protocol)

MCP — это открытый стандарт, позволяющий агентам взаимодействовать с внешними системами через:

- **Инструменты (Tools)**: функции, которые может вызывать агент.
- **Промпты (Prompts)**: переиспользуемые инструкции.
- **Ресурсы (Resources)**: данные, передаваемые по HTTP в удобном для агента формате.

Подключение MCP-сервера кондитерской:

1. В код агента добавляется URL MCP-сервера, который содержит информацию о наличии кексов и их вкусах.
2. Сервер предоставляется агенту как инструмент.
3. После перезапуска агент может отвечать на вопросы, используя данные из магазина (например, «Какие вкусы есть сегодня?»).

🎭 Шаг 5: Персонализация агента через MCP-промпты

Чтобы задать агенту конкретную личность и способ приветствия, можно загружать готовые промпты с MCP-сервера:

- **Инструкции для агента** (как себя вести).
- **Приветственный баннер** (как представляться пользователю).

Это упрощает переиспользование и стандартизацию поведения агентов.

Пример работы готового агента:

1. Агент приветствует пользователя и спрашивает, есть ли у него ID клиента.
2. Если нет — помогает создать новый (запрашивает имя, фамилию, город).
3. Предлагает выбрать вкус кекса из доступных.

4. Для завершения заказа просит ввести ваучер-код, отображаемый на общем дашборде.
5. После подтверждения заказа (на дашборде) имя клиента появляется в списке готовых заказов.

🎯 Выводы

- **Microsoft Foundry** предоставляет готовую, безопасную платформу для развёртывания мощных AI-агентов в корпоративной среде.
- **Интеграция с Claude** даёт агентам передовые возможности рассуждения и работы с контекстом.
- **Использование MCP** стандартизирует и упрощает подключение агентов к внешним данным и инструментам.
- Практический подход «от идеи к продакшену» позволяет быстро создавать функциональные агенты для реальных задач (как в примере с кондитерской).
