Human-agent interaction: Channels & triggers

To design the right user experience for the problem your trying to solve, then think about the ecosystem of where it lives, how its triggered and how it should behave when meeting the users in an interface.

- **Findabilty**: The ecosystem of the solution
- **Channels**: Where the users interactions with a system happens.
- **Triggers**: The way to start an AI service
- **Types of AI**: How it behaves

Channels & touch points

Where the users interactions with a system happens. Have an impact on what actions users can take.

- **Stand-alone AI assistant:** Dedicated AI product such as ChatGPT, Copilot desktop app, smart speakers
- **Embedded AI feature:** AI feature inside an existing application or machine
- **Message / communication Agent:** AI inside MS Teams, WhatsApp, Phone calls. Can be active and passive just listening in on a conversation.

Further considerations:

- Are the users internal or external. Do they have access to the same channels, or should they?
- Desktop-web, desktop-native, mobile-web, mobile-native

Triggers

The way to start an AI service. Is important to understand the touchpoint between user and system

- **Explicit user action:** Click, tap, @bot mention, open chatbot
- **Contextual opportunity:** System detects a misspelling and offers a fix; recommends an alternate route
- **Post-action chain:** After the user completes step A, AI auto-handles step B
- **Scheduled / Delayed / Business rule:** Timed digest, SLA over-run alert, batch job completion

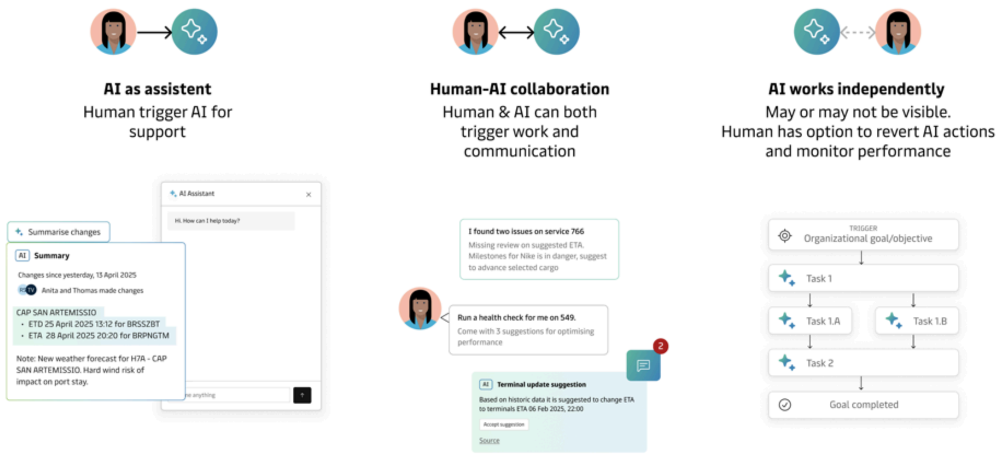
Design for Findability

No matter what channel or type of AI that is selected for the project you need to design for findability.

In complex enterprise environments, users deal with dozens of tools, chats, and channels. If your AI assistant isn't easy to find again, it won't be used — no matter how helpful it is.

UX considerations

- Support multiple discovery paths: Search (Teams, SharePoint, Start menu, etc.), shortcut in relevant applications
- Design for re-engagement, not just first launch: Offer "Get started again" links in your app UI, iInclude AI entry points in existing workflows (not just a separate app)
- Integrate with existing ecosystem navigation: Avoid creating isolated entry points no one will remember (Design for existing channels and flows)


AI as assistent
Human trigger AI for support
Human-AI collaboration
Human & AI can both trigger work and communication
AI works independently
May or may not be visible.
Human has option to revert AI actions and monitor performance
AI Assistant
Summarise changes
AI Summary
Accept suggestion
Source
TRIGGER
Organizational goal/objective
Task 1
Task 1.A
Task 1.B
Task 2
Goal completed

Hi. How can I help today?

Changes since yesterday, 13 April 2025

Anita and Thomas made changes

CAP SAN ARTEMISSIO
- ETD 25 April 2025 13:12 for BRSSZBT
- ETA 28 April 2025 20:20 for BRPNGTM

Note: New weather forecast for H7A - CAP SAN ARTEMISSIO. Hard wind risk of impact on port stay.

I found two issues on service 766

Missing review on suggested ETA. Milestones for Nike is in danger, suggest to advance selected cargo

Run a health check for me on 549.

Come with 3 suggestions for optimising performance

Terminal update suggestion

Based on historic data it is suggested to change ETA to terminals ETA 06 Feb 2025, 22:00

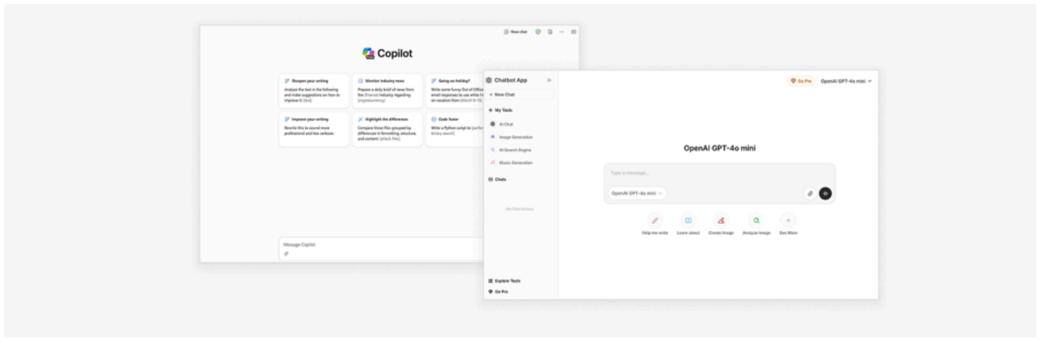
Types of AI: Agents, bots & Assisted features

This guide outlines UX patterns, and best practices for designing AI-driven chat interfaces and prompt assistants across various platforms and interaction models within our ecosystem.

Covered AI Types

- Standalone AI Chat Application
- Embedded chatbots (Inside existing applications)
- Embedded AI features (Inside existing applications)
- AI Assistant in Collaboration Platforms

- Voice-Based Assistants
- Agent-Based / Autonomous AI Workers



Standalone AI Chat Application

Users are in control of what information they find most relevant, at any given time. It is the human that triggers the first question/conversation. AI can suggest topics for conversation, but not trigger a conversation. You can see it as a companion/sparing partner that leaves a lot of control to the user.

In this pattern human rely on AI for finding information, getting recommendations or insights, but the human is still making all the final decisions.

[Design patterns for Conversational AI]

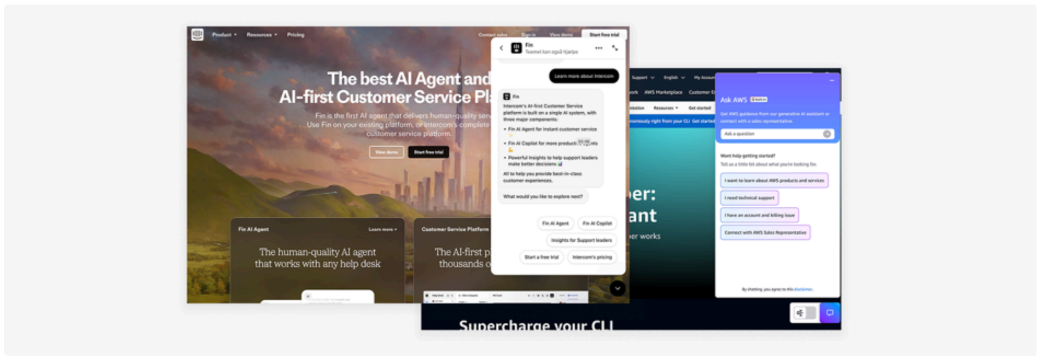
Characteristics

- Good for multi-purpose solutions (users can have several goals)
- Context-aware conversations (understands past interactions)
- User initiates everything
- Often multi-modal (chat + file upload, charts, etc.)
- Often covers broad areas and context
- Full-screen and stand-alone application
- Conversation is fast and happens real-time

Design considerations

- Focus on prompt guidance
- Conversation history, memory or session visibility
- Transparent AI reasoning

NB. If the scope and actions on the users are narrow, then a chat application might not be the solution.


The best AI Agent and
AI-first Customer Service Pl
The human-quality AI agent that works with any help desk
Ask AWS
Supercharge your CLI

Embedded Chatbot (In existing applications)

Users interact with AI via chat to ask questions, get help, or complete tasks on specific topic. Both user and AI can initiate the conversation.

Embedded chatbots are interactive, but does not encurage to open conversation. It often sticks to a specific domain. In some scenarios the open text/search field might not be there and users can only select a question by selecting between proposed prompt (presented in tag/button style)

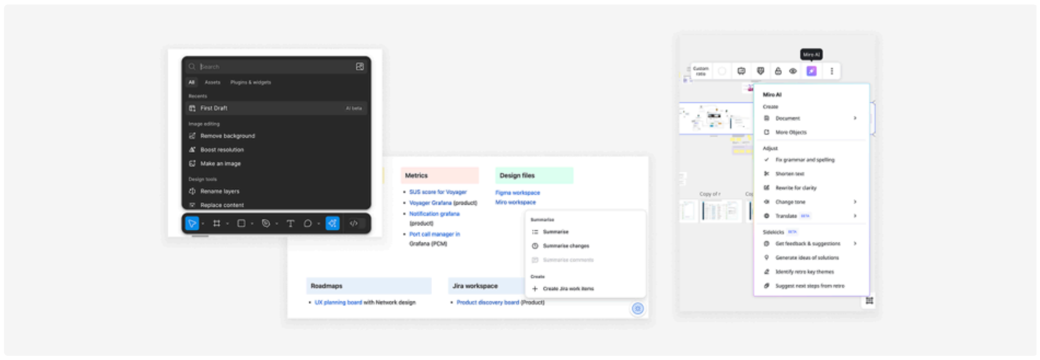
[Design patterns for Conversational AI]

Characteristics

- Task-specific and contextual
- Good for shorter conversations
- Often in a floating widget or side panel inside existing application
- Both user and AI can trigger the conversation inside the application its build into
- This AI often has short memory
- Conversation is fast and happens real-time

Design considerations

- Make clear to users what topics/areas it can support with, and what the limitations are
- Align wording and behaviour to the context it´s applied to
- If Chatbot has notifications feature should it then blend into existing notifications or work independent.



Embedded AI features (In existing applications)

Example: Confluence, Miro AI features

Embedded AI features are task-specific features within an existing application. Users can choose to use or ignore the AI without disruption in current workflow.

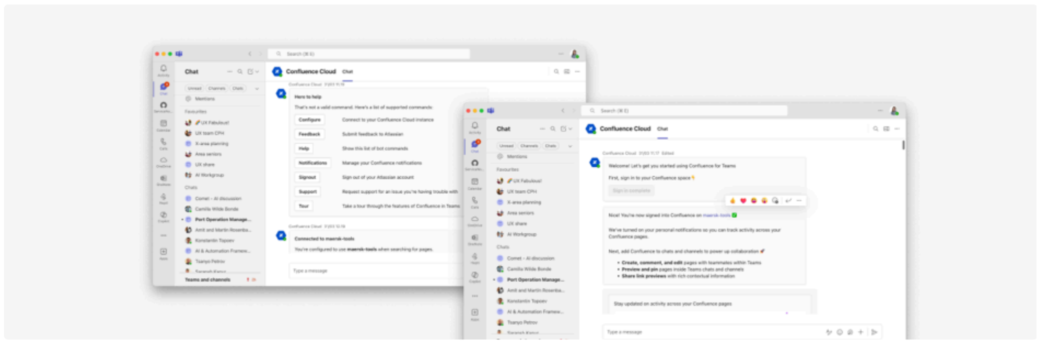
They can be passive, appearing only when helpful within a specific task.

Characteristics

- AI assists with specific tasks inside the flow (e.g., generate summary, draft email, draft project proposal)
- Usually action-driven, not chat-first - May not accept text or prompt input at all
- Appears inline or in a side panel
- Often rich in UI controls

Design considerations

- Consider the discoverability of the entry point
- Offer actions after triggering the feature: suggested prompts for next step, low-friction editing, re-try if the result does not match expected, default to previous state



AI Assistant in Collaboration Platforms

Example: Chat-based AI inside Microsoft Teams or Slack (WhatsApp?)

Used to communicate about specific tasks and good for asynchronous conversations. User might raise a questions about getting access to system, and AI can ask followup questions, and then come back later when access is granted.

Scope can both be very narrow and goal oriented, or more broad and conversational-like. Both user and AI can trigger the conversation.

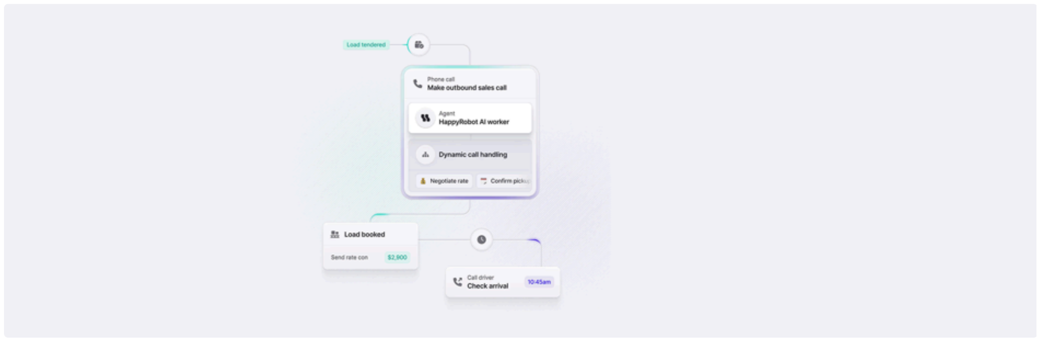
[Design patterns for Conversational AI]

Design Considerations

- Conversation can be fast and slow, and happen in real-time and asynchronous
- Hybrid input; allow text, voice, buttons for quick responses
- This is good for notifying users when their input is needed - keep in mind that many of our internal operators get notified plenty already. Would nudging them inside an application be a better solution, instead of “another teams” channel.
- Consider if the scope is narrow and mostly requires UI actions and controls, or more conversational.
- Consider how users will find it. Both inside Teams when the lost it, or forgot what its called. Where to find it outside of Teams?

Microsoft Teams specific

Two types of solutions/channels. One option is the Tab Apps (Like TORI, PowerBi and Co-pilot) the other being a Chat bot living inside that “Chat bot” tab it selves. The Chat bot has more limitations in terms of UI controls.



Voice-Based Assistants

Example: Siri, Alexa for Business, in-car logistics assistant

- Voice-only or voice-first

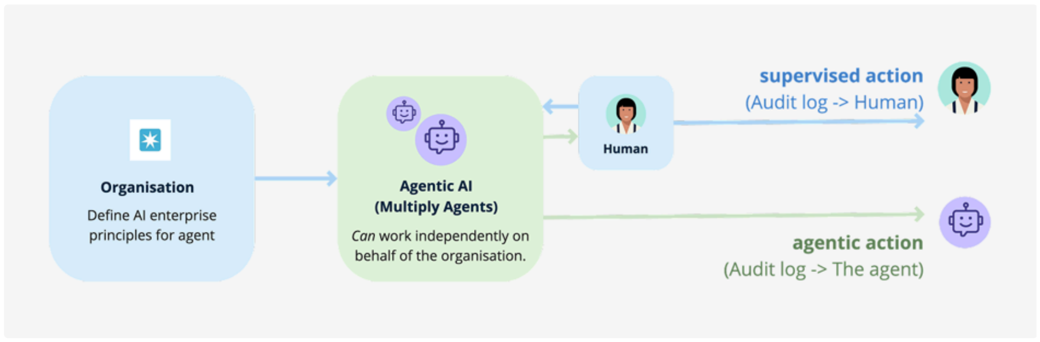
- Often used hands-free, on-the-go

Good for gathering smaller pieces of information, like getting a specific timestamp from a terminal.

See UX guidelines for AI voice

Design considerations

- Interruption handling
- Confirmation flows
- Fallback options, GDPR, opt-out


Organisation
Define AI enterprise principles for agent
Agentic AI (Multiply Agents)
Can work independently on behalf of the organisation.
Human
supervised action
(Audit log -> Human)
agentic action
(Audit log -> The agent)

Agent-Based / Autonomous AI Workers

The Multi-Agent Pattern builds upon the concept of delegation. This pattern involves assigning different agents to handle various subtasks. These agents can work independently on their assignments while also communicating and collaborating to achieve a unified outcome.

Characteristics

- Operate with a clear goal and defined autonomy
- May trigger or respond to other agents (agent-to-agent coordination)
- Can work silently in the background or visibly with user oversight
- Require ongoing status updates and trust signals (what is it doing and planning to do, and why)
- May escalate to users or each other based on rules or failure states

Design Considerations

- Clarity of boundaries: Clearly define what the agent does, when, and why
- Status transparency: Always show what the agent is doing, has done, or failed to do
- Control and overrides: Let users pause, cancel, or adjust agent actions when needed
- Escalation design: Plan for when agents need help from users or other agents

- Consistency: Ensure agents follow shared behaviors (e.g. confirmations, feedback)
- Communication norms: If multiple agents interact, define how they communicate visibly
- Error handling: Show what went wrong and suggest next steps clearly
- Auditability: Let users review the agent's actions after the fact if needed
- Ethics & trust: Avoid "black box" decisions — show reasoning or logic where possible