

[Open in app](#) ↗[Sign up](#)[Sign in](#)**Medium** Search

PART II

Creating Voice Interaction Flows

UX Design for voice interfaces

**Poulami Chakraborty** · Follow

Published in UX Collective

6 min read · Dec 31, 2018



Listen



Share

Photo by [Kelly Sikkema](#) on [Unsplash](#)

This is the second of a 3-part series on my design process for Voice Interfaces. The first part covered defining the product. This part is about designing the conversation.

Voice interfaces are now implicitly expected to be conversational — as opposed to command-based modality and/or one-turn exchanges.

Designing a conversational interface involves influencing user requests to, and modeling the responses from our voice agent.

A voice input or request has two layers:

Utterance: What the user says is a user utterance. Basically, an utterance is how a user phrases a request.

Example: A user can frame a request in any of the following ways:

“Set an alarm for 8'o clock”

“Alarm 8 a.m”

“I want to set an alarm”

“Wake me up at 8”

These are just a few. There may be 1000 ways to ask to set an alarm.

While it is not possible to list the exhaustive set of utterances for any given goal, listing a few variations *would be helpful in later part of the process.*

Intent: This is the main objective behind any user request.

Example: In the above cases, the intent is to set an alarm.

Intent can be again divided into two types: primary intent is the actual task the user wants to do.

Secondary intents are for tasks to complement **primary** tasks.

Example: If the primary intent is to order a coffee, choosing a delivery address different from the default one could be a secondary intent.

Our assistant will support all primary user intents which correspond to the goals identified in the previous step.

(Any intent mentioned from here on will refer to a primary intent)

Conversation Modelling — Happy Flow

For a particular intent, we first model a complete conversation for the ‘happy paths’. ‘Happy paths’ are ones in which the user reaches the ideal outcome in the least (or within a certain) number of steps.

This step also helps us recognize the ‘slots’ or ‘variables’ required to fulfill the request. **Variables** are inputs or additional information which the user needs to provide to complete the intent.

Example:

*User: I'd want to order a **Salted Caramel Mocha Frappuccino***

Assistant: Sure, what size would you like?

*User: **Grande***

*Assistant: 1 Grande Salted Caramel Mocha Frappuccino. Would you want it delivered to your **office**?*

User: Yes

Assistant: Ok. Should I order it?

*User: Wait, I would like **non-fat milk***

Assistant: Sure. Changed your milk preference to non-fat milk. Should I confirm the order?

User: Yes

Assistant: Order placed successfully. Your coffee will be reach you soon.

This is one possible way by which a user can complete a coffee order. Users may provide different amounts of information, and in different ways. Some may give all the information up front, while others provide it piece by piece. Some may have a lot of information to add, others just the mandatory information.

To handle this, the assistant needs to know the **mandatory variables** in what order to ask for them to complete the flow (in the above example in bold). These variables are necessary to complete the request.

The assistant needs to be, also, able to understand the **optional variables** (in the above example in bold italics). These are filters/customizations or choices to override defaults, which users may request.

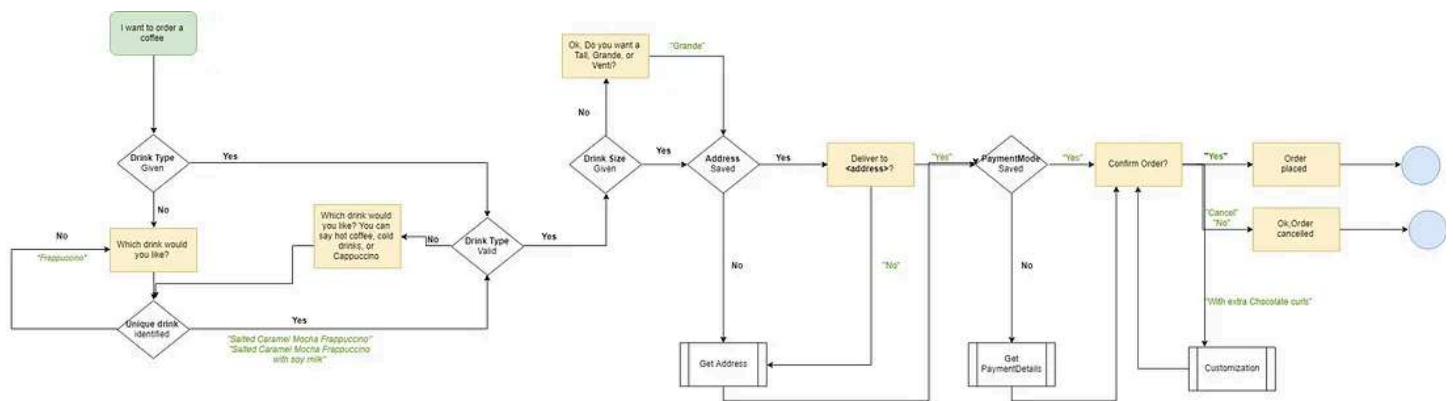
At the end of this step, we list out these mandatory variables, and all the optional variables we can think of, before we proceed to the next step.

The catalogue of utterance variation, *carried out earlier*, increases the chance of stumbling upon all possible **variables** in a flow.

User flow

Now it is time to extensively chart out all possible 'flows' or paths to complete a goal. It takes into account all the ways that a user may choose to phrase a request.

Prompts will be designed to ask for any missing mandatory variable & get the user to goal completion through the most optimized path.



Sample flow

The flow needs to handle edge cases and error cases.

Example of possibilities to consider when creating flows:

Invalid requests — User asks for something which is not possible. (ex: “Rate Ready Player Five 6 stars on Goodreads” or “the fifth one” while choosing from a list of 4 items)

Details in Multiple results — How much information to give out in case of multiple results (ex: “What do I have planned today”? — if there are 5 results, how do you read it out: summarize, read out 1 in detail and wait for user to say “next”, or read out the title of each)

Disambiguation — Some user requests may have ambiguous variables and require a little clarification before they can be fulfilled. Policies need to be defined for such cases. (ex: if user says weather in Springfield, you may ask each the user to clarify of the 34 Springfields they mean, or choose one based on their past behavior, or which is nearest)

Empty result state — We cannot forget about the ‘0 results’ found pages. Without any specifically added responses for those cases, the default dialogues may end up sounding very mechanical (ex: “Find me non-stop AA flights from San Francisco to Vancouver” — there are no matching results. The assistant shouldn’t respond with “I found 0 results”).

No user response — Sometimes, the user may fail to answer within the response time after a prompt. There’s no way of knowing why a user didn’t reply, but repeating the same prompt again may not be the best available option. What should be the behavior for a particular state - Can we consider providing hints as to what they can say? Ask for a different piece of information, or phrase the earlier question in a different way? Not do anything?

Over-answers — User may provide more information than what is being prompted.

Of course, it depends on the platform capability, but the best experience is for the platform to be able to accommodate such cases. Asking for information which user has already provided makes the system look stupid. (ex:

User: Set me a reminder

Assistant: What is this reminder for?

User: “Doctor appointment for 4 p.m. tomorrow

User has already provided the time for the appointment. So, it's better not to prompt them for it again)

Optional variables — While user shouldn't be prompted for optional variables, they shouldn't be prevented from providing them either. So, the VUI should make accommodations to support such requests. In other words, we need to make sure that the conversation shouldn't fall apart (“I am sorry, I don't understand that) when user says “with extra Chocolate curls” instead of “yes”/ “no”.

Intelligent Learning (over time)—Are there any prompts that can be removed over time as the system learns more about the user? (“Book me an Uber” — if the user regularly requests Uber at 5 p.m. from his office location, maybe we can avoid asking the user for destination under those circumstances; inform the user of the assumption directly in the confirmation state. However, user should be easily able to override such defaults.)

Context-retention—Natural conversations are not always one-turn. We need to consider how user may follow up after any response. (ex: “Is it going to rain today?” followed by “What about weekend?”)

No hierarchy — Unlike GUIs, there are no back buttons in voice interfaces.

Conversations only move forward. The user should be able to provide additional (optional) information at any stage without losing any previous progress. (Example: In the above happy path, modifying the type of milk during confirmation did not require the user to provide delivery address again)

Feedback of cancellation — Conversations don't end abruptly. So, it's necessary that we make sure to account for cases where user cancels a flow, or doesn't opt for any suggested action. The assistant should provide the proper feedback that the task has been cancelled.

No assistant response — In some cases like “Play music”, “Turn on lights”, assistant needn't give any reply, and just complete the task. These kinds of design specifications should also be included in the VUI.

Sample dialogue

Once we have finalized the flow, we can phrase the assistant responses or **dialogues** to make your interactions feel natural. Moreover, we need to adjust the language and tone to align with the persona we have defined for the assistant.

At this step, it helps to go back to the utterance variation catalogue and test if the dialogues seem ‘natural’ with any user utterance. If not, some response variation needs to be added & mapped to different user utterances.

Example: will the assistant response work for both “I’d want to order a Salted Caramel Mocha Frappuccino” and “Is it possible to get a Salted Caramel Mocha Frappuccino delivered now?”

Variations of responses are added so that the responses don’t feel repetitive and robotic.

[Artificial Intelligence](#)[Chatbot](#)[Voice Assistant](#)[UX](#)[User Experience](#)



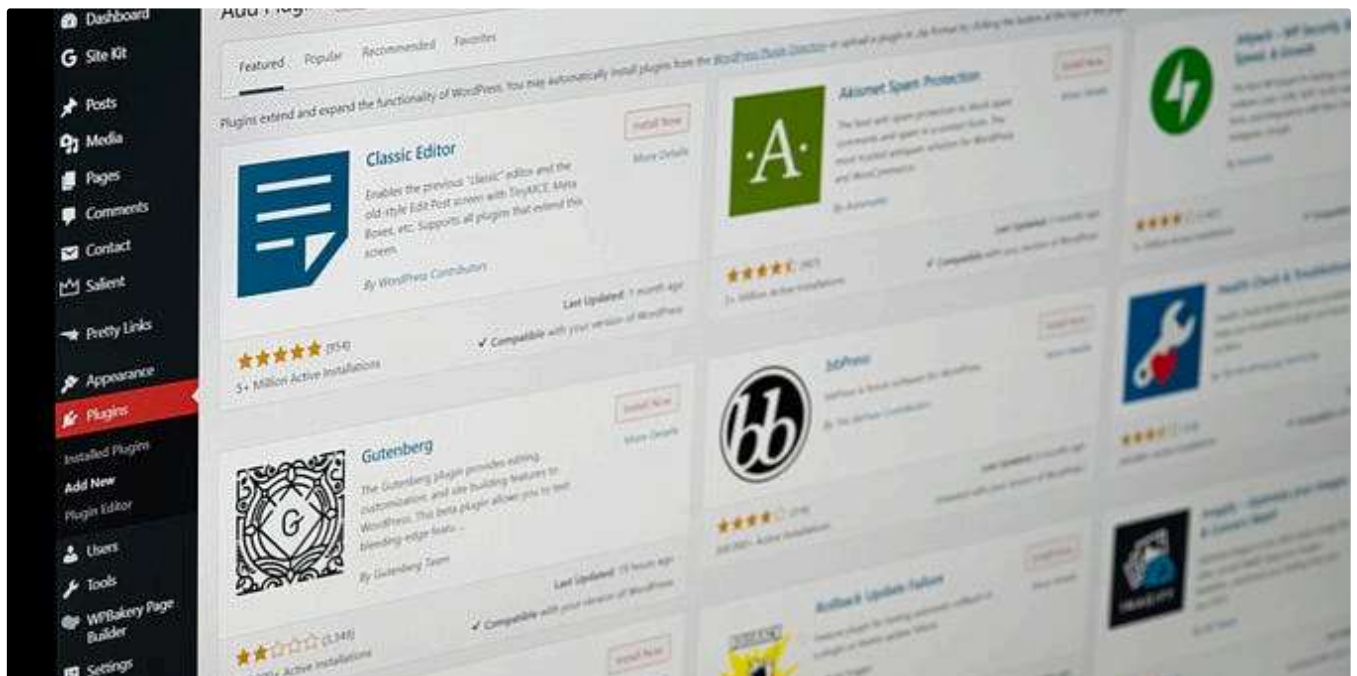
Follow

Written by Poulami Chakraborty

96 Followers · Writer for UX Collective

Voice and Product Designer documenting thoughts and learnings.

More from Poulami Chakraborty and UX Collective



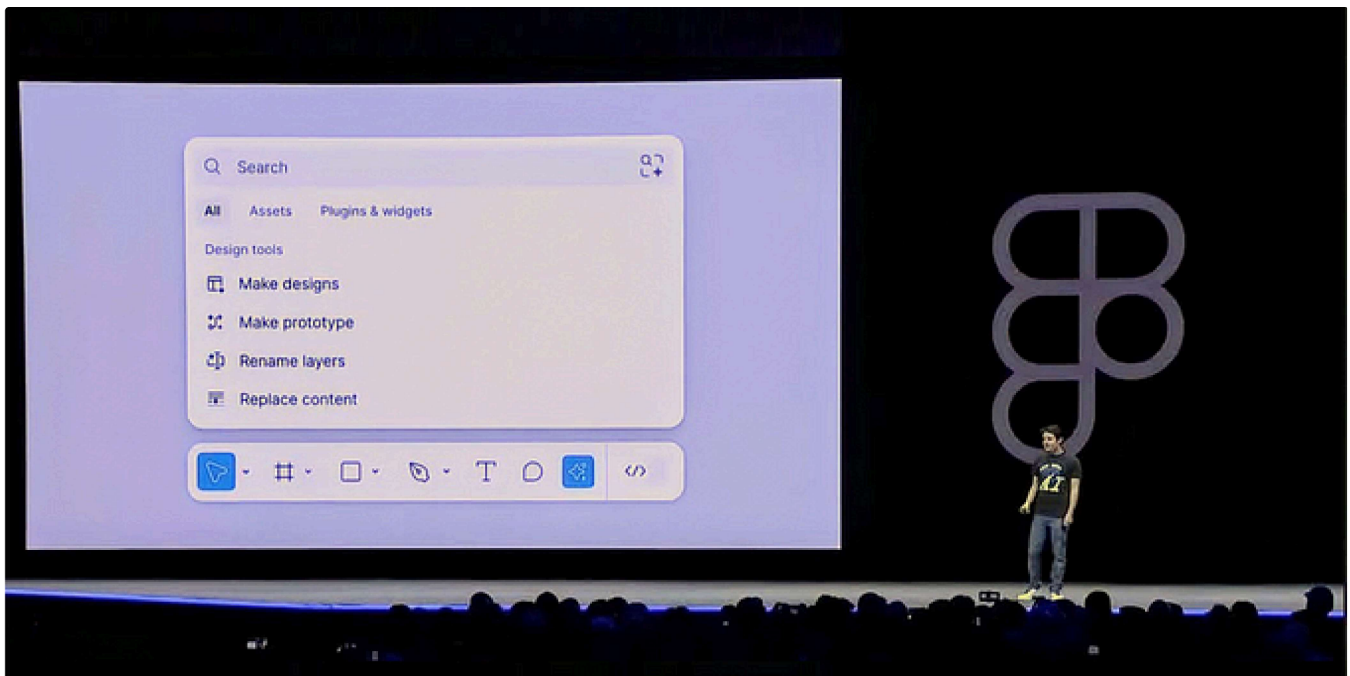
 Poulami Chakraborty in UX Collective

Designing for information density

Balancing visual and structural simplicity

Jul 17, 2021  242





Darren Yeo in UX Collective

Figma AI is going to change your design job applications

And it starts with your design portfolios.



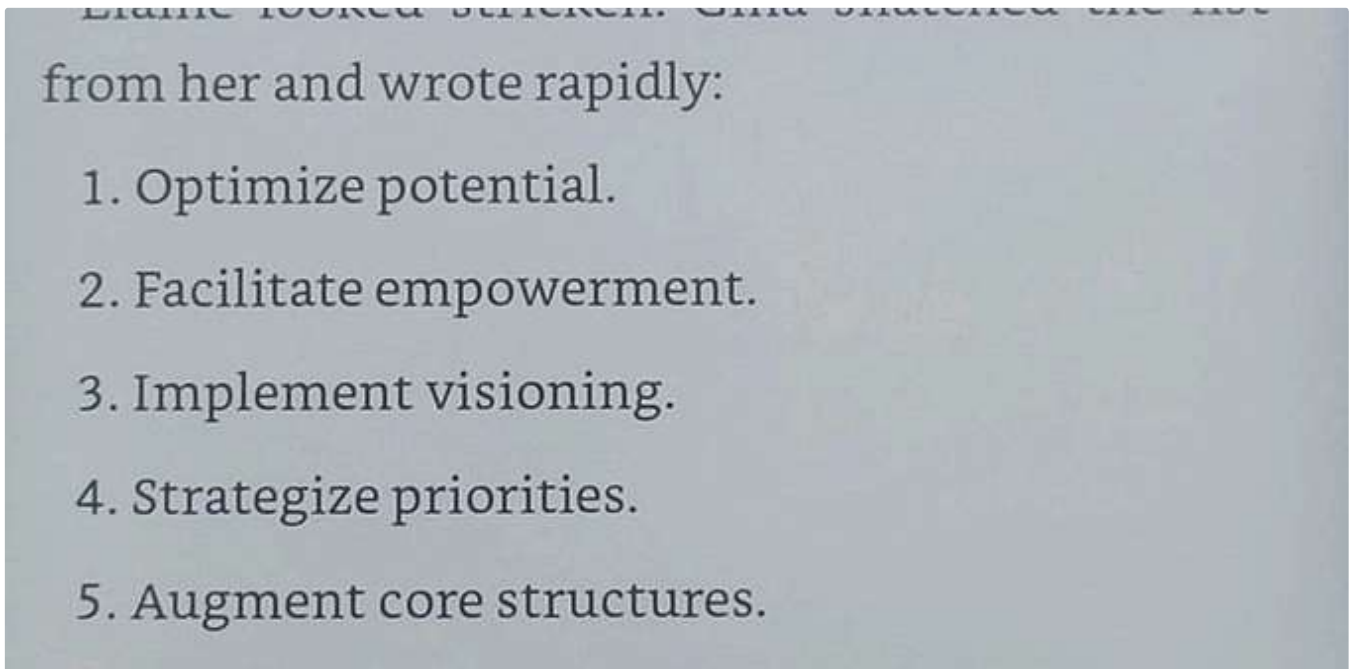
Jul 12



700



7


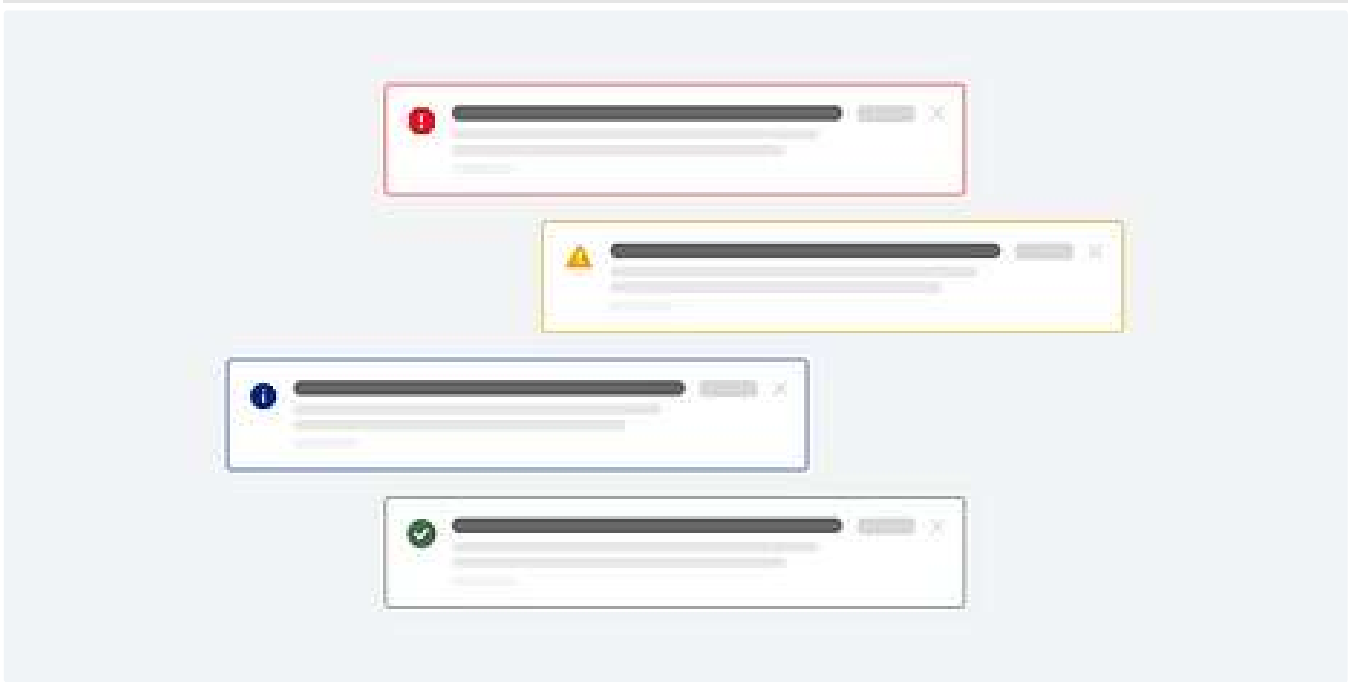


Pavel Samsonov in UX Collective

Nike's \$25B blunder shows us the limits of "data-driven"

The illusion of data as "objective" conceals that it rarely shows you the whole picture. Making decisions based on the easiest data to...

Aug 2 🖱️ 1.7K 💬 32

 Poulami Chakraborty in UX Collective

Thinking in frameworks: communicating system status

Notifications, alerts, and indicators.

Aug 24, 2021 🖱️ 228

[See all from Poulami Chakraborty](#)[See all from UX Collective](#)

Recommended from Medium



LucianoSphere (Luciano Abriata, PhD) in Towards Data Science

Web Speech API: What Works, What Doesn't, and How to Improve It by Linking It to a GPT Language...

Part of a series on how modern AI and other technologies could assist more efficient human-computer interactions



Dec 6, 2023



561



7



Nandini Stocker

To ask or not to ask, that is the question.

Considerations for deciding on a command or conversational interaction model for a voice tech solution

★ Oct 3, 2022 🖱 113 💬 1

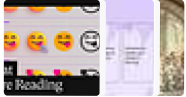


Lists



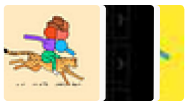
Interesting Design Topics

257 stories · 700 saves



Stories to Help You Grow as a Designer

11 stories · 875 saves



Figma 101


7 stories · 671 saves



Icon Design

36 stories · 377 saves



 Paulo Taylor

Improving Real-Time Voice Interfaces with Advanced Text-to-Speech Technology

Speed is a critical factor, especially for real-time voice interfaces like those used in the mobile app Voice Over Your Phone (Voyp)

May 20

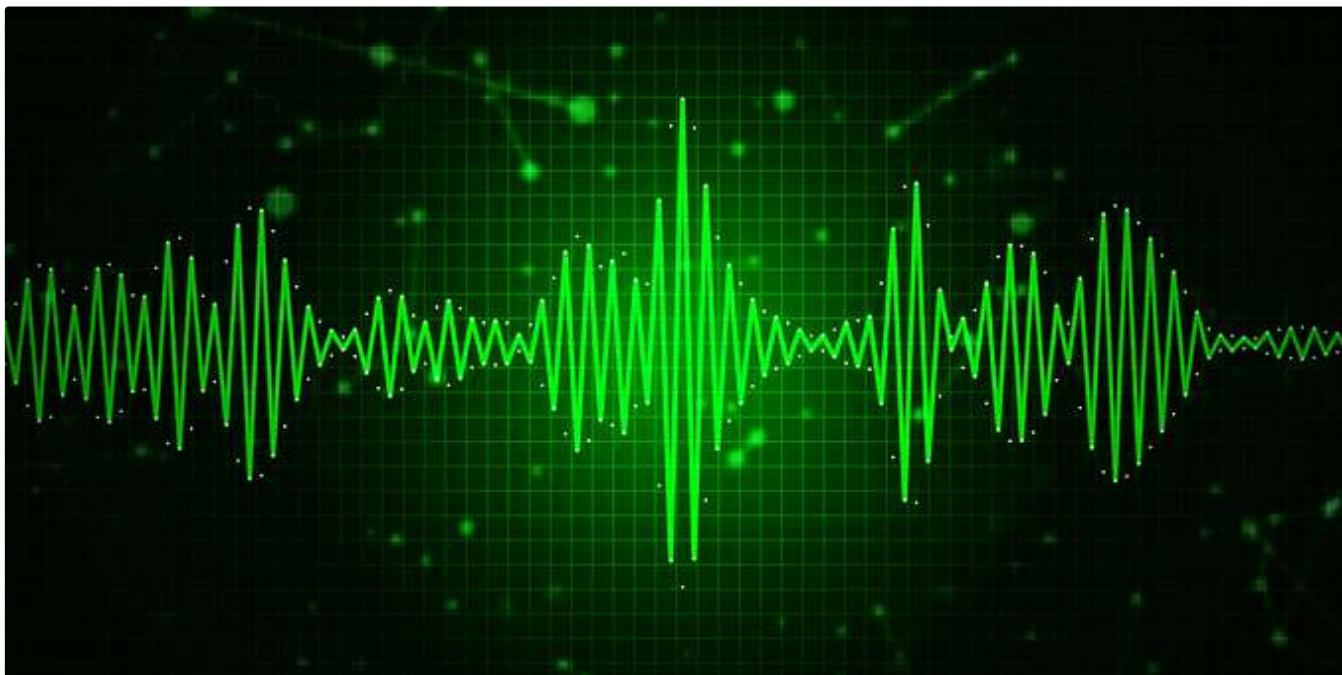


 Kshitij Agrawal

The next big AI-UX trend — it's not Conversational UI

Everything is an input and everything is an output. What if you could browse ALL your things in ONE fluid interface?

★ Apr 2 🤝 316 💬 6



 Atul Kumar

Building a Streaming Speech-to-Text Application with FastAPI and Amazon Transcribe

In this tutorial, we'll walk through building a streaming speech-to-text application using FastAPI and Amazon Transcribe. We'll cover the...

Jul 7 🤝 5 💬 2





Mark O'Brien

Whisper & Python for Video Transcription

Explore the Whisper Model from OpenAI for audio transcription made easy, fully local



Mar 4



9



See more recommendations