

ABC:

Why Use AI-Powered WhatsApp Chatbots for Sexual and Reproductive Health Services

This guide helps IPPF Member Associations decide, design, and implement an AI-powered WhatsApp chatbot to strengthen initial guidance for sexual and reproductive health services.

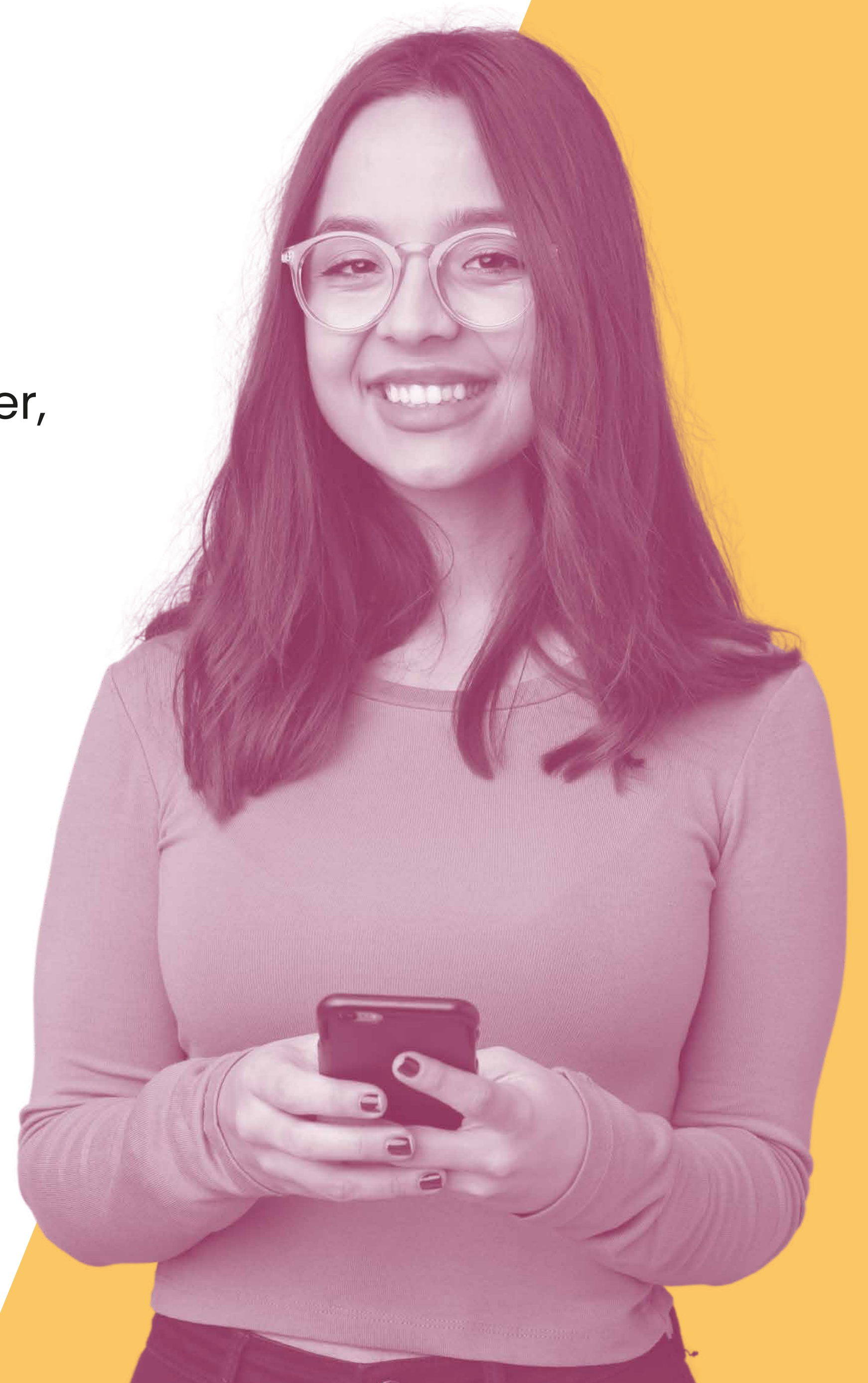
What is a chatbot?

A tool that helps provide better guidance, respond faster, and support users without replacing human care.

A chatbot can be useful when there is:

- High demand for frequently asked questions.
- Long waiting times.
- A need for clear and reliable information.
- Demand for privacy and confidentiality.
- A need for timely referral to human support.

The organization's need should be addressed first, and only then select the tool



What Type of Solution Does the Organization Need?

Not every WhatsApp channel needs AI

Three types of solutions can coexist on WhatsApp:

1

Human-agent support

Ideal for sensitive cases, complex decisions, emotional support, and situations outside the scope of automation.

2

Rules-based automated navigation

Useful for organizing simple pathways: menus, buttons, appointment booking, initial classification, or closed-ended questions.

3

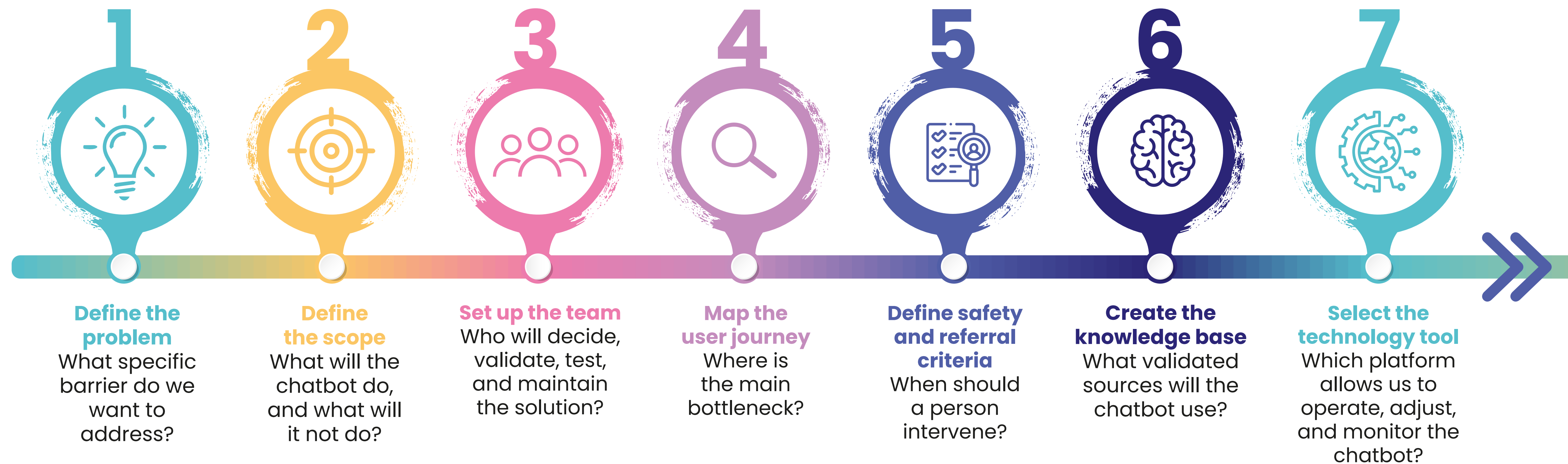
AI-powered chatbot

Allows users to ask questions in natural language and receive answers from an organization - validated knowledge base.

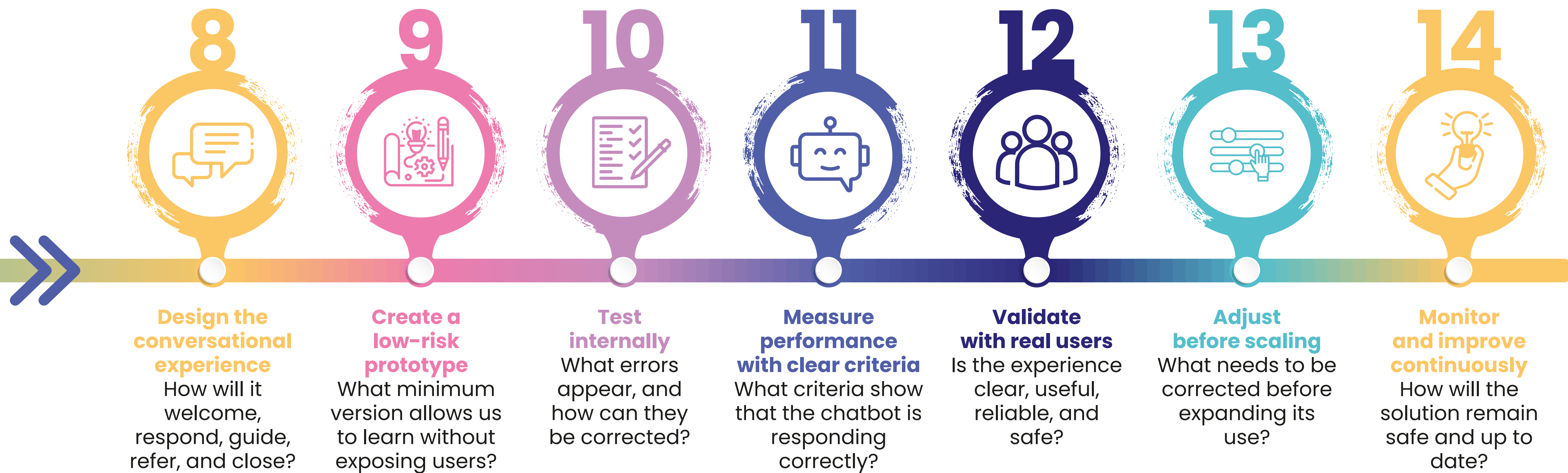
The best option is often a hybrid model: rules to organize + AI to guide + people to provide support.



14 Steps to Implement an AI-Powered WhatsApp Chatbot



14 Steps to Implement an AI-Powered WhatsApp Chatbot



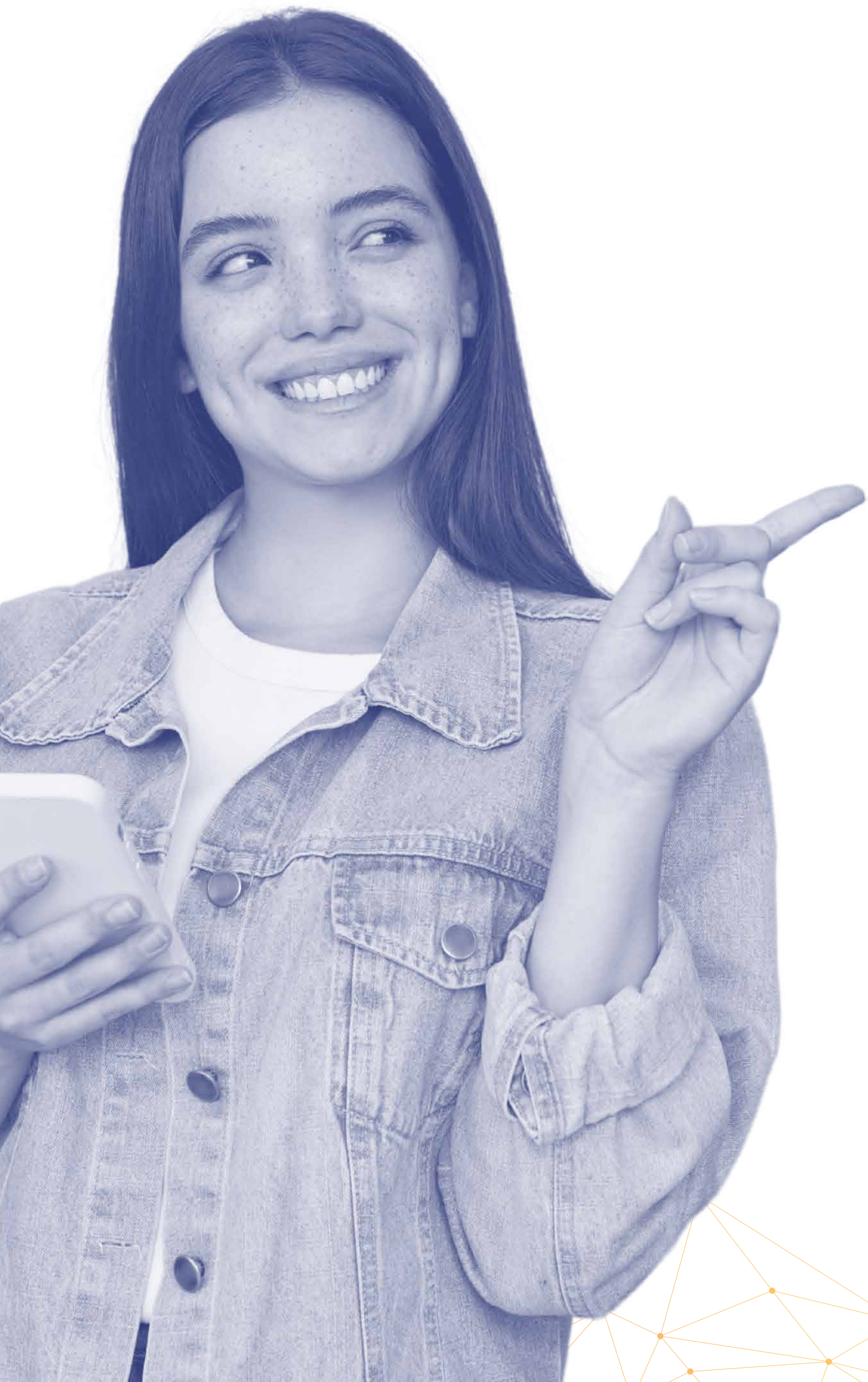
From Profamilia's Experience to a Replicable Model

The value lies not in copying the solution, but in adapting its lessons.

Profamilia developed an AI-powered chatbot to strengthen the first point of contact on WhatsApp for its **Mía** service. The objective was to reduce information barriers, improve response speed, and complement human agents' work.

mía
Mi cuerpo, mi autonomía.





Transferable lessons

1 Start with the problem, not with AI.

2 Limit the use case.

3 Design with boundaries and human oversight.

4 Test, measure, and adjust.

5 Identify errors.

6 Validate with users.

7 Consider sustained maintenance over time.

8 Continuously validate the final version.

