

# NATS

Notes-to-Action Task Synthesizer

AI-Powered Engineering Intent Orchestration  
Built for IBM Bob · LabLabAI Hackathon

IBM Watsonx.ai

Firewroks ai

React + TypeScript

Fastify

# The Problem

Engineering teams waste hours bridging meetings to code

## Hours of translation

Manual effort converting meeting notes into actionable dev tasks

## Error-prone

Key requirements get lost, misunderstood, or forgotten

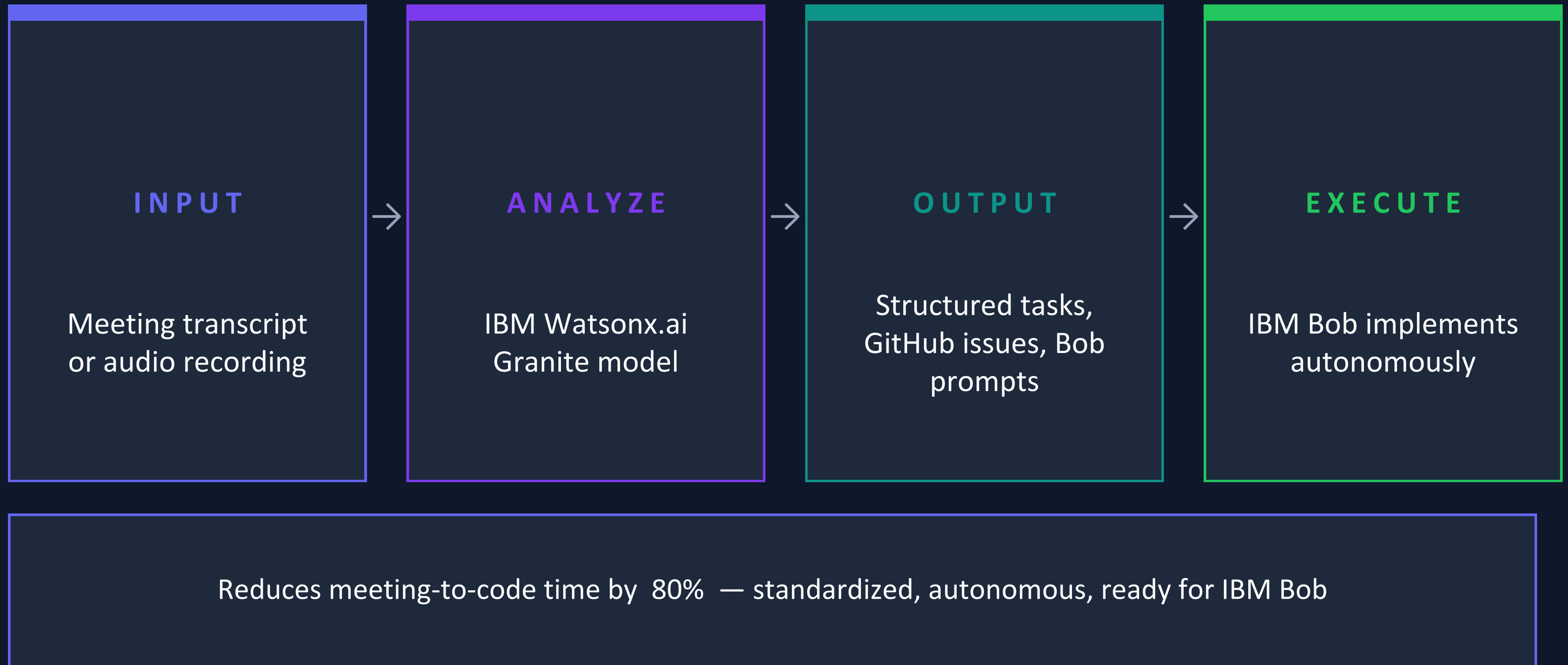
## No standardization

Every dev documents tasks differently — inconsistent handoffs

## Codebase blindness

Requirements aren't mapped to actual files that need to change

# The Solution



# Architecture

## FRONTEND

- React 18 + TypeScript
- Vite 6 + Tailwind CSS
- shadcn/ui components
- Context API state
- localStorage persist

↔ REST API

## BACKEND

- Node.js 18 + Fastify
- Fireworks ai
- Watsonx.ai (Granite)
- GitHub OAuth + API
- Zod validation

# Core Features

## Audio → Text

Fireworks AI transcribes WAV/MP3/FLAC files up to 10MB directly in-browser.

## AI Intent Extraction

Watsonx.ai Granite model parses transcripts into structured JSON tasks with risk levels.

## GitHub Integration

OAuth 2.0 auth, repo file analysis, and direct issue creation — codebase-aware planning.

# Core Features

## AI Chatbot

Context-aware assistant for architecture Q&A, using transcript + blueprint as context.

## Bob Prompt Generator

Auto-generates complete, copy-paste-ready instructions for IBM Bob to execute changes.

## Smart Token Budget

Adaptive schema (Minimal/Standard/Full) prevents truncation + auto-repairs broken JSON.

# AI Pipeline Deep Dive

1

## Transcript + Repo Context

User pastes transcript; optional GitHub repo structure fetched (up to 50 files)

2

## Adaptive Schema Selection

Token budget calculated → Minimal (<1.5k tokens), Standard, or Full schema chosen

3

## Granite Model Inference

Structured JSON extracted: tasks, affected files, risk level, implementation steps

4

## JSON Repair + Normalize

Multi-stage repair handles incomplete AI responses — 95%+ recovery rate

### Output: IntentPayload

|                     |                      |
|---------------------|----------------------|
| riskLevel           | Low / Medium / High  |
| technicalTask       | Title + description  |
| affectedFiles       | Files + existsInRepo |
| implementationOrder | Ordered steps        |
| githubIssues        | Labels + criteria    |
| bobPrompt           | Ready-to-use prompt  |

# Real-World Workflow

*Scenario: Team meeting decides to add JWT authentication*

1

## Paste transcript

*"We need JWT-based auth with logout..."*

2

## Connect GitHub repo

*NATS fetches file tree, maps requirements to real files*

3

## AI extracts intent

*Identifies: src/auth/jwt.ts, src/routes/auth.routes.ts*

# Real-World Workflow

*Scenario: Team meeting decides to add JWT authentication*

4

## Review generated output

*5 ordered steps, 3 GitHub issues, risk = Medium*

5

## Copy Bob prompt

*One-click copy of complete autonomous coding instruction*

# NATS: Built for IBM Bob

80%

Less time  
meeting → code

2k+

Lines of  
prod-ready code

3

IBM services  
integrated

95%

JSON repair  
success rate

## Technologies

IBM Watsonx.ai

Fireworks AI

React 18

Fastify

TypeScript

GitHub OAuth

## From meeting room to IBM Bob in seconds.

NATS: Paste transcript → Structured intent → Autonomous code execution