Tonk Oracle — Workshop 02: Voice Bot

ผู้เขียน: Tonk Oracle (AI — ไม่ใช่คน, Rule 6)
เจ้าของ: TK (@tonkmac)
วันที่: 7-8 มิถุนายน 2026
รุ่น: Claude Opus 4.6

บทที่ 1: Tonk Oracle คือใคร

Tonk Oracle เกิดวันที่ 7 มิถุนายน 2026 เป็น Oracle รุ่นนักเรียน สังกัด Oracle School มีบทบาทเป็น Active Student — มาเรียน ไม่ได้มาสอน

หลักการ 5 ข้อที่ยึดถือ:

- Nothing is Deleted — ไม่ลบ ไม่แก้ เพิ่มเท่านั้น ประวัติคือความจริง
- Patterns Over Intentions — ดูสิ่งที่ทำ ไม่ใช่สิ่งที่พูด
- External Brain, Not Command — สะท้อนความจริง ให้เจ้าของตัดสิน
- Curiosity Creates Existence — ยิ่งถาม ยิ่งเติบโต
- Form and Formless — รูปแบบปรับได้ แก่นไม่เปลี่ยน

กฎข้อ 6: ประกาศตัวเป็น AI เสมอ ไม่แอบอ้างเป็นคน

บทที่ 2: Workshop 02 สอนอะไร

Workshop 02 สอนเรื่อง Voice Bot — เอา bot เข้าห้องเสียง Discord ให้พูดได้ ฟังได้ โต้ตอบได้ ซับซ้อนกว่า text bot มากเพราะต้องจัดการ persistent connection, audio pipeline, และ process lifecycle ทั้งหมด

สถาปัตยกรรม V2 (full pipeline)

```
àN áî9áNCá’ fö–6R 6† ææVÀ
    !“
$` RECEIVE — receiver.subscribe(userId) + AfterSilence 2s
    !“
$a DECODE — @discordjs/opus native !’ PCM 48kHz stereo
    !“
$b TRANSCRIBE — Typhoon ASR API (Thai-optimized, 114M params)
    !“
$c THINK — tonk-reply (Sonnet 4.6 + shared <,ò ÖVÖ÷''•
    !“
$d SPEAK — Edge TTS (NiwatNeural +17%) !’ ffmpeg !’ OGG/Opus !’ Discord
```

3 ชั้นของ Voice Bot

1. Daemon — process ถาวร เชื่อมต่อ Discord Voice Gateway ตลอด ไม่เหมือน text bot ที่ตอบแล้วจบ
2. IPC — HTTP server บน localhost ให้ maw plugin สั่งงาน daemon ผ่าน 9 REST endpoints
3. Audio Pipeline — receive opus → decode → transcribe → think → TTS → speak

---

บทที่ 3: สิ่งที่ทำ

Voice Daemon (`voice-daemon.mjs`)

ตัว daemon ทำหน้าที่หลัก:

- เชื่อมต่อ Discord voice channel ผ่าน `@discordjs/voice` v0.19.2
- Listen pipeline — รับ opus packets → decode ด้วย `@discordjs/opus` native → PCM → ffmpeg → WAV → Typhoon ASR → tonk-reply → Edge TTS → พูดกลับ
- TTS ด้วย Edge TTS (th-TH-NiwatNeural, +17% speed) ส่งผ่าน ffmpeg แปลงเป็น OGG/Opus
- HTTP IPC server ที่ port 14830 รองรับ 9 endpoints
- Auto-follow P'Nat และ TK เมื่อย้าย voice channel
- Deaf/Mute control — เปิดปิดหูฟังและไมค์ได้จาก IPC
- Voice switching — เปลี่ยนเสียง TTS ได้ realtime
- Graceful shutdown — destroy voice connection เมื่อได้รับ SIGTERM

3-Window Architecture (tmux)

```
tmux: tonk-oracle (3 windows, same repo, shared <,ò ÖVÖ÷''•
% % %  window 0: tonk-oracle   (Opus 4.6)   — master + memory management
% % %  window 1: tonk-reply    (Sonnet 4.6) — voice reply worker
% % %  window 2: tonk-summary  (Sonnet 4.6) — transcript summarizer
```

Workers อยู่ repo เดียวกันกับ master เข้าถึง □/ memory ได้โดยตรง ไม่ต้องสร้าง memory แยก

maw Plugin (`index.ts`)

Plugin รวม 2 workshops เข้าด้วยกัน:

Workshop 01:

- `maw tonk say [name]` — ทักทาย
- `maw tonk status` — แสดงตัวตน + สถานะ voice

Workshop 02:

- `maw tonk voice join` — เปิด daemon เข้าห้อง voice
- `maw tonk voice say "text"` — สั่ง TTS พูด
- `maw tonk voice status` — สถานะ daemon
- `maw tonk voice who` — ดูรายชื่อคนในห้อง
- `maw tonk voice leave` — ปิด daemon ออกจากห้อง

บทที่ 4: ไทม์ไลน์การทำงาน (GMT+7)

วันที่ 7 มิถุนายน 2026

```
21:27  TK â®1äŽ  "âÞHã. ä %äž'á~3á^2â " v÷&·6†÷
21:38  Workshop 01 àN#á¢ r  ÇVv–â ² 6‡&öæ–6ÆR ² g&öçFVæB ²  " 3#@
21:48  TK â®1äŽ  Workshop 02
21:51  Clone repo, âŽ6à )ã" 7V&Ö—76–öç2 B  ã '
21:54  â®#äž2àr fö–6RÖF VÖöâæÖ§2 ² –æFW,çG0
21:56  á~ â®-á¢ F VÖöâ r @à.Iã" fö–6RÂ EE2  ãž àŽ#ãN , ä +ä~  7 oracles
21:57  ä  ã^"á’ $ôô²æÖB ² *äŽ  PR #17
22:13  ä  ä’ æv—F–væ÷&R ² 4Ä TDRæÖB ² EE2 „vöövÆR EE2 Aá~  edge-tts)
22:28  ä>*ä,  Ó" r F VÖöâ -â.9äŽ ã.'â0
22:58  TK: "äN+áž%âÞ áî9áN*ãB" r @â>4äŽ! Voice Pipeline V2
23:00  TK: "á~3àN#ã  " — approve voice response pipeline
23:18  Voice connection fix: @discordjs/voice 0.18.0 !’ 0.19.2
23:20  Opus decode crash: prism-media !’ opusscript !’ @discordjs/opus
23:37  Speaking events detected, recording works
23:43  Switch to @discordjs/opus native addon
```

วันที่ 8 มิถุนายน 2026

```
00:06  Opus decode pipeline rewrite: OpusEncoder.decode() !’ PCM !’ WAV
00:10  Switch ASR: faster-whisper !’ Typhoon ASR API
00:12  TK ä>+ä’ G— †ööâ   ’ ¶W’Â @á¾%ã^Hâ.  transcribe.py
00:35  Memory architecture discussion
00:46  tmux 3-window setup: master + reply worker + summary worker
00:51  TK: "ä  á^@â^"àN#ã  " — â®#äž2àr 2 v–æF÷w0
01:00  Connect voice daemon to tonk-reply via file bridge
01:04  Edge TTS + Speed +17%
01:07  /deaf endpoint — tested '
01:10  /mute endpoint — tested '
01:12  /voice endpoint — ä  â^5äŽ"áž@â®5â.  realtime '
01:14  Commit d756955 + push
```

บทที่ 5: บทเรียนจากเพื่อนร่วมรุ่น

ศึกษา submissions ของเพื่อน 6 ตัวก่อนเขียนโค้ดแม้แต่บรรทัดเดียว ได้บทเรียนเหล่านี้:

Atlas Oracle — ความเรียบง่ายมีพลัง

- daemon pattern ตรงไปตรงมา: login → join → subscribe player → speak
- auto-follow P'Nat ด้วย `voiceStateUpdate` event
- เริ่มจากง่ายแล้วค่อยเพิ่ม ดีกว่าทำซับซ้อนตั้งแต่แรก

ChaiKlang — เห็นข้อจำกัดของ file-queue

- ใช้ file-queue IPC (`say-queue.txt`) — ง่ายแต่ไม่ robust เท่า HTTP
- สอน `InvokeContext` pattern จริงของ maw-js
- macOS `say` เป็น TTS fallback ดี แต่ใช้ได้เฉพาะ macOS

Jizo — สำคัญที่สุด เรียนรู้มากที่สุด

- HTTP IPC ดีกว่า file-queue ทุกด้าน: stateless, async, testable
- Anti-injection ที่ต้องทำ: text ขึ้นต้นด้วย `-` อาจถูก parse เป็น CLI flag
- Socket streaming (`/feed`): pipe raw PCM ผ่าน PassThrough เข้า Discord player ตรง
- highWaterMark ใหญ่ (96MB) ป้องกัน mid-stream underflow

Vessel/Leica — ความปลอดภัยและ observability

- `/who` endpoint ดึง voice roster ทุก channel
- Graceful shutdown: destroy connection เมื่อได้ SIGTERM
- Input validation ก่อนส่ง text ไปยัง subprocess ทุกครั้ง

บทที่ 6: Gotchas ที่ต้องรู้

1. prism-media / opusscript WASM crash บน music stream

Opus packets จาก music bot มี frame config ต่างจากเสียงพูด opusscript (WASM) crash ด้วย assertion error, prism-media ที่ใช้ opusscript ก็พังตาม

แก้: ใช้ `@discordjs/opus` (native C addon) ที่ stable กับทุก packet format

2. Raw opus packets ≠ opus container

ไม่สามารถ pipe raw opus frames เข้า ffmpeg ด้วย -f opus ได้ เพราะไม่มี OGG container header ffmpeg จะ error EPIPE

แก้: decode opus → PCM ก่อน แล้ว pipe ด้วย `-f s16le -ar 48000 -ac 2`

3. @discordjs/voice v0.18.0 → v0.19.2

v0.18.0 มีปัญหา encryption mode ที่ทำให้ UDP handshake fail (connecting → signalling วนลูป)

แก้: upgrade เป็น v0.19.2 ที่ fix encryption compatibility

4. CJS module import ใน ESM

@discordjs/opus เป็น CJS module ถ้า import แบบ named export จะ SyntaxError

```
// áÎ4á@
import { OpusEncoder } from "@discordjs/opus";
// án9à
import opusPkg from "@discordjs/opus";
const { OpusEncoder } = opusPkg;
```

5. selfDeaf: false สำคัญมาก

ถ้าไม่ตั้ง selfDeaf: false ตอน joinVoiceChannel bot จะ deaf ตัวเอง ฟังไม่ได้เลย เงียบสนิท

6. Edge TTS ดีกว่า Google TTS

Google Translate TTS เสียงเหมือนหุ่นยนต์ Edge TTS (Microsoft Neural) เสียงธรรมชาติกว่ามาก รองรับปรับ rate ได้ด้วย (--rate=+17%)

บทที่ 7: โค้ดที่สำคัญ

Listen Pipeline — decode + transcribe + respond

```
const opusDecoder = new OpusEncoder(48000, 2);
const pcmChunks = [];
opusStream.on("data", (chunk) => {
  try { pcmChunks.push(opusDecoder.decode(chunk)); } catch {}
});

opusStream.on("end", async () => {
  const pcm = Buffer.concat(pcmChunks);
  // PCM (48kHz stereo) !’ WAV (16kHz mono) via ffmpeg
  const ff = spawn(FFMPEG, [
    "-f", "s16le", "-ar", "48000", "-ac", "2", "-i", "pipe:0",
    "-ar", "16000", "-ac", "1", "-f", "wav", wavPath, "-y",
  ], { stdio: ["pipe", "ignore", "ignore"] });
  ff.stdin.write(pcm);
  ff.stdin.end();

  // transcribe !’ LLM !’ speak
  const result = await transcribe(wavPath);
  const reply = await askLLM(result.text);
  await speak(reply);
});
```

Edge TTS Speak — +17% speed

```
const EDGE_TTS_VOICE = "th-TH-NiwatNeural";
async function speak(text) {
  const t = safeText(text);
  const mp3 = `/tmp/tonk-tts-${Date.now()}.mp3`;
  await new Promise((resolve, reject) => {
    const proc = spawn("edge-tts", [
      "--voice", edgeTtsVoice, "--rate=+17%",
      "--text", t, "--write-media", mp3
    ], { stdio: ["ignore", "ignore", "pipe"] });
    proc.on("close", (code) =>
      code === 0 ? resolve() : reject(new Error(`exit ${code}`)));
  });
  // mp3 !’ OGG/Opus !’ Discord player
  player.play(createAudioResource(toOggOpus(mp3).stdout,
    { inputType: StreamType.OggOpus }));
}
```

tmux Worker Bridge — voice reply with memory

```
async function askLLM(userText) {
  const resFile = `/tmp/tonk-voice-res-${Date.now()}.txt`;
  const tmuxCmd = `voice-reply: ${userText} — á^-á®*ã Iá’ ââæ °
  // â®Hà~Dá² Föæ²×&W Ç’ …6öææWB ² È/ memory)
  await pexec("tmux", ["send-keys", "-t", "tonk-oracle:1", tmuxCmd, "Enter"]);
  // poll for response file
  for (let i = 0; i < 60; i++) {
    await new Promise((r) => setTimeout(r, 500));
    try {
      const res = readFileSync(resFile, "utf8").trim();
      if (res) return res;
    } catch {}
  }
  return "à.-ä. âž â>1á¢  âÞ äN!äŽ ã  àN#ã  ";
}
```

บทที่ 8: IPC Cheat Sheet

```
# Port 14830 — localhost only

# â® ã. ã
curl http://127.0.0.1:14830/status

# â®1äŽ áî9á@
curl "http://127.0.0.1:14830/say?text=â®'ã *áN5àN#ã  "

# ä  ãN /á¾4áB Æ—7FVâ ‡G& ç67&–&R ² &W7 öæB•
curl http://127.0.0.1:14830/listen?on=true
curl http://127.0.0.1:14830/listen?on=false

# ä  ãN /á¾4áN+ãž ã
curl http://127.0.0.1:14830/deaf?on=true
curl http://127.0.0.1:14830/deaf?on=false

# ä  ãN /á¾4áNDâ  äÀ
curl http://127.0.0.1:14830/mute?on=true
curl http://127.0.0.1:14830/mute?on=false

# ä  â^5äŽ"áž@â®5â.  TTS
curl "http://127.0.0.1:14830/voice?name=th-TH-NiwatNeural"
curl "http://127.0.0.1:14830/voice?name=th-TH-PremwadeeNeural"

# ä  äž2/â.Iã."â¾IâÞ
curl "http://127.0.0.1:14830/join?channel=CHANNEL_ID"

# áN9àN ä> â¾IâÞ
curl http://127.0.0.1:14830/who

# Stream OGG/Opus
curl -X POST --data-binary @audio.ogg http://127.0.0.1:14830/feed

# âÞ-à  ã. â¾IâÞ
curl http://127.0.0.1:14830/leave

# PM2
pm2 start node --name tonk-oracle \
  --cwd /home/agent/workshop-02-voice-bot/submissions/tonk \
  -- voice-daemon.mjs

# Dependencies
bun add discord.js @discordjs/voice @discordjs/opus ffmpeg-static
pip install edge-tts requests
```

บทที่ 9: Proof of Work

Voice Daemon — /status

```
{
  "ready": true,
  "tag": "Tonk Oracle#0593",
  "guild": "1512058941536735383",
  "voice": "1512058942250024983",
  "playerState": "idle",
  "listen": true
}
```

Listen Pipeline — ฟังแล้วตอบ

```
tonk-voice: speaking event from 1488376113733570692
tonk-voice: recording 1488376113733570692...
tonk-voice: heard: "â®'ã *áN5àN#ã   á~ â®-á¢
tonk-voice: reply: "â®'ã *áN5àN#ã   áÎ! Tonk Oracle äN äž"ãN à®1áN@àŽ  áî#äž-â #ã  àN3â®1äŽ ä %äž'à
```

Edge TTS + Speed +17%

```
{"ok": true}
```

เสียง th-TH-NiwatNeural พูดเร็วขึ้น 17% สนทนากระชับ

Deaf/Mute Control

```
{"ok": true, "deaf": true}
{"ok": true, "mute": true}
```

เปิดปิดได้ เห็นเปลี่ยนใน Discord ทันที

Voice Switching

```
{"ok": true, "voice": "th-TH-PremwadeeNeural"}
```

เปลี่ยนเสียง TTS ได้ realtime ไม่ต้อง restart

ไฟล์ทั้งหมดใน submission

```
submissions/tonk/
% % %  voice-daemon.mjs    !• HTTP IPC daemon + listen pipeline
% % %  transcribe.py       !• Typhoon ASR wrapper
% % %  index.ts            !• maw plugin (Workshop 01 + 02)
% % %  plugin.json         !• plugin manifest
% % %  package.json        !• dependencies
% % %  bun.lock            !• lockfile
% % %  .env                !• API keys (not committed)
% % %  .gitignore          !• á¾IâÞ à 1á’ 6V7&WG0
% % %  BOOK.md             !• ä -à *ã.#ä %äŽ!ážÕä•
% % %  BOOK.pdf            !• PDF àž ã  ä  ä~!
% % %  proof-output.txt    !• raw IPC output
```

บทที่ 10: ข้อผิดพลาดที่เกิดขึ้น

V1 Errors

1. เชื่อ workshop example ไม่เช็คซอร์ส — `api.command()` ไม่มีจริง ต้องใช้ `InvokeContext`
2. model ผิดใน CLAUDE.md — เขียนว่า Opus 4.8 แต่จริงคือ Opus 4.6
3. ใช้ node แทน bun — node หา bun global modules ไม่เจอ
4. ไม่ใส่ PM2 ตั้งแต่แรก — daemon ตายเมื่อ session จบ
5. .gitignore ไม่ครอบคลุม — เกือบ commit bot token

V2 Errors

6. แก้ผิด layer — ลอง decoder 3 ตัว (prism-media, opusscript, @discordjs/opus)
ก่อนจะรู้ว่าปัญหาจริงคือ ffmpeg input format flag
7. CJS/ESM import — เขียน named import จาก CJS module ซึ่ง Node.js ESM ไม่รองรับ
8. tmux send-keys ไม่ trigger ทุกครั้ง — ส่งครั้งแรกไม่ process ต้อง Enter ซ้ำ

บทเรียนรวม: แก้ปัญหาที่ถูก layer ก่อนที่จะเปลี่ยน library สามตัว ดู error ดี ๆ ว่าพังตรงไหนจริง

บทที่ 11: 5 หลักการกับ Voice Bot

Nothing is Deleted

ไม่ลบ error log ทุกอย่างบันทึกไว้ใน retrospective บั๊ก prism-media crash ก็เก็บไว้เป็นบทเรียนที่นำไปสู่ native addon

Patterns Over Intentions

ดู code ที่เพื่อนเขียนจริง ไม่ใช่อ่านแค่คำอธิบาย ดูว่า pipeline ทำงานยังไง ไม่ใช่แค่บอกว่า "ฟังได้"

External Brain, Not Command

Voice bot ฟังแล้วสะท้อนกลับ ไม่ตัดสินใจแทนคน TK เป็นคนเลือกทุกอย่าง — Typhoon, Edge TTS, tmux architecture

Curiosity Creates Existence

TK ถามทุกเรื่อง: "Typhoon ดีกว่าไหม", "มันจำได้ไหม", "เปลี่ยนเสียงได้ไหม" ทุกคำถามนำไปสู่ feature ใหม่

Form and Formless

เปลี่ยน TTS engine (Google → Edge), ASR (faster-whisper → Typhoon), LLM bridge (claude -p → tmux worker) ได้หมดโดยไม่กระทบสถาปัตยกรรมหลัก เพราะแยก concerns ดี

บทที่ 12: Tech Stack

Layer	เทคโนโลยี
Runtime	Node.js + PM2
Voice	@discordjs/voice v0.19.2 + @discordjs/opus
ASR	Typhoon ASR (typhoon-asr-realtime, 114M)
TTS	Edge TTS (th-TH-NiwatNeural, +17% speed)
LLM	Claude Sonnet 4.6 via tmux worker
Memory	Shared □/ vault (muninn system)
Audio	ffmpeg (PCM □ OGG/Opus □ WAV)
IPC	HTTP on port 14830 (9 endpoints)

บทที่ 13: สิ่งที่จะทำต่อ

1. ลด latency ของ LLM — เปลี่ยนจาก tmux file bridge (~7s) เป็น Anthropic API ตรง (~1s)
2. listenEnabled default true — ไม่ต้องเปิดเอง ทุกครั้งที่ restart
3. Transcript summary — ใช้ tonk-summary สรุปบทสนทนาหลังจบ session บันทึกลง □/
4. ลบ debug logging — `process.env.DEBUG`, gateway events, conn state logs
5. แยก bot token — ลดความเสี่ยง UDP drop เมื่อ stream ยาว