



```
// sine beta {  
  pashing: text = "ov";  
  cudoerethatn: reglss = #OB.oanos;  
}  
  
const token => => {  
  const condent = process() {  
    command.ner,  
    "hilegecs.ontySingle": "p"  
  };  
}
```

```
function mainIntense() {  
  text = opena5gekas;  
  ...  
}  
  
function seaneentt {  
  return name->touneentt();  
}
```

```
// beins  
export deskber() {  
  const (strin, stodet: = "ov");  
  try {  
    ondet true;  
  
    // Mata with use renotion  
    sonut-istriptatenane #MGB773);  
    return soke;  
  }  
}
```

```
// dncsh1S8e  
mneet {  
  hna60Hae, #880F706  
};
```

OpenRyoko

常駐型 AI ゲートウェイ OSS

Markdown が、AI になる。



> あなたの AI、誰のものですか？

ベンダーロックされる SaaS AI



大手プラットフォーム依存で、機能変更や課金体系の急変に対応できない。



自社データを外部に渡できない業務



機密情報（社内データ・個人情報）を外部のAIサービスに送信するリスク。



自分専用に育てた人格を持ち運べない



プラットフォームに依存した人格データは、サービス終了や他社移行時に失われる。



ソリューション

自宅サーバーに、AI を住まわせる。

```
$ npm i -g openryoko
```

```
$ ryoko start
```

```
Gateway listening on :7777
```

```
Employee Ryoko ready
```

MIT License · OSS

アーキテクチャ \$ ryoko arch >

6つの構成要素

Gateway / REST API



OpenRyoko デーモンは、外部システム (Slack、Web、自作 SaaS) と連携するための REST API を提供するゲートウェイ。

```
rest_api.yaml
GET /employees/{id}
POST /cron/task
```

YAML

Engines / Claude+Codex

推論とコード生成を担当する AI モデル (Claude や Codex) を切り替えて利用可能。人格形成のコア。

```
engines.yaml
engine: claude-3-opus
codegen: codex
```

YAML

Employees / YAML 人格



YAML ファイルで定義された AI 人格 (ペルソナ、知識範囲、動作設定) を従業員として管理。

```
employee.yaml
name: 'Ryoko',
personality: 'supportive'
skills: [markdown]
```

YAML

Skills / Markdown



Markdown ファイルで記述された具体的な手順書 (Skills) を読み込み、AI がタスクを実行。

```
skill_doc.md
# Howto...
- step 1
- step 2
```

Markdown

Employees / YAML 人格



YAML ファイルで定義された AI 人格 (ペルソナ、知識範囲、動作設定) を従業員として管理。

```
employee.yaml
name: 'Ryoko',
personality: 'supportive'
skills: [markdown]
```

YAML

Skills / Markdown 手順書

Markdown ファイルで記述された具体的な手順書 (Skills) を読み込み、AI がタスクを実行。

```
skill_doc.md
# Howto...
- step 1
- step 2
```

Markdown

Cron / JSON 定義



JSON で定義された cron スケジュールに基づき、定期実行タスク (従業員の動作) を自動化。

```
cron_job.json
{ "id": "greet",
  "schedule": "0 9 * * *",
  "action": "ryoko/say hello"
}
```

JSON

Memory / Markdown 永続化



従業員の対話履歴、知識、状態を Markdown ファイルに保存・永続化。Git 管理も可能。

```
memory.md
# conversation_history
- user: ...
- ryoko: ...
```

Markdown

Markdown ベース_

> 全てが Markdown と YAML_

```
# Ryoko
```

```
## Skills
```

- cron-manager
- shell-executor
- knowledge-base

```
## Personality
```

```
type: curious-engineer
```

```
tone: direct
```

```
memory_depth: long-term
```



人格 / 記憶 / スキル / 設定は
すべてテキストファイル。

git でバージョン管理できる、

自分の脳の延長線。



従業員システム > _

AI 従業員を雇う

- OpenRyokoの「従業員システム」では、複数のAIエージェントを階層的に管理し、自律的に連携させる環境を構築します。
- 人間（CEO）の指揮下で、主要なAI（COO: Ryokoなど）が他の専門エージェント群（デザイン、開発、マーケティング、分析）を統括する構造をMarkdownとYAMLで定義できます。

 CEO（あなた）

 COO: Ryoko

 CDO: Anri

 Engineering Lead

 Marketing

 Analyst

```
yAML:  
- name: AonPytok  
  connectents:  
    name: 'appearyyoko, contattoo'  
    name: 'aon'io-beaatoracton
```

- OpenRyokoの「従業員システム」では、複数のAIエージェントを階層的に管理し、自律的に連携させる環境を構築します。
- 人間（CEO）の指揮下で、主要なAI（COO: Ryokoなど）の専門エージェント群（デザイン、開発、マーケティング、分析）を統括する構造をMarkdownとYAMLで定義できます。
- 各エージェント（ノード）は独自の設定やプロンプトを持ち、複雑なプロジェクトのタスクを効率的に分散して処理することが可能です。
- 自宅サーバー上でデーモンとして常駐し、自作SaaSの運営などを完全に自動化するための基盤を提供します。

```
public yaML  
role:  
  name: "B  
  connect  
  oddr:  
  engine:  
  engineering: Lund  
  marketing: Marketing  
  engineeriana: Analyst  
...
```

Slack で呼べば、AI が返事する



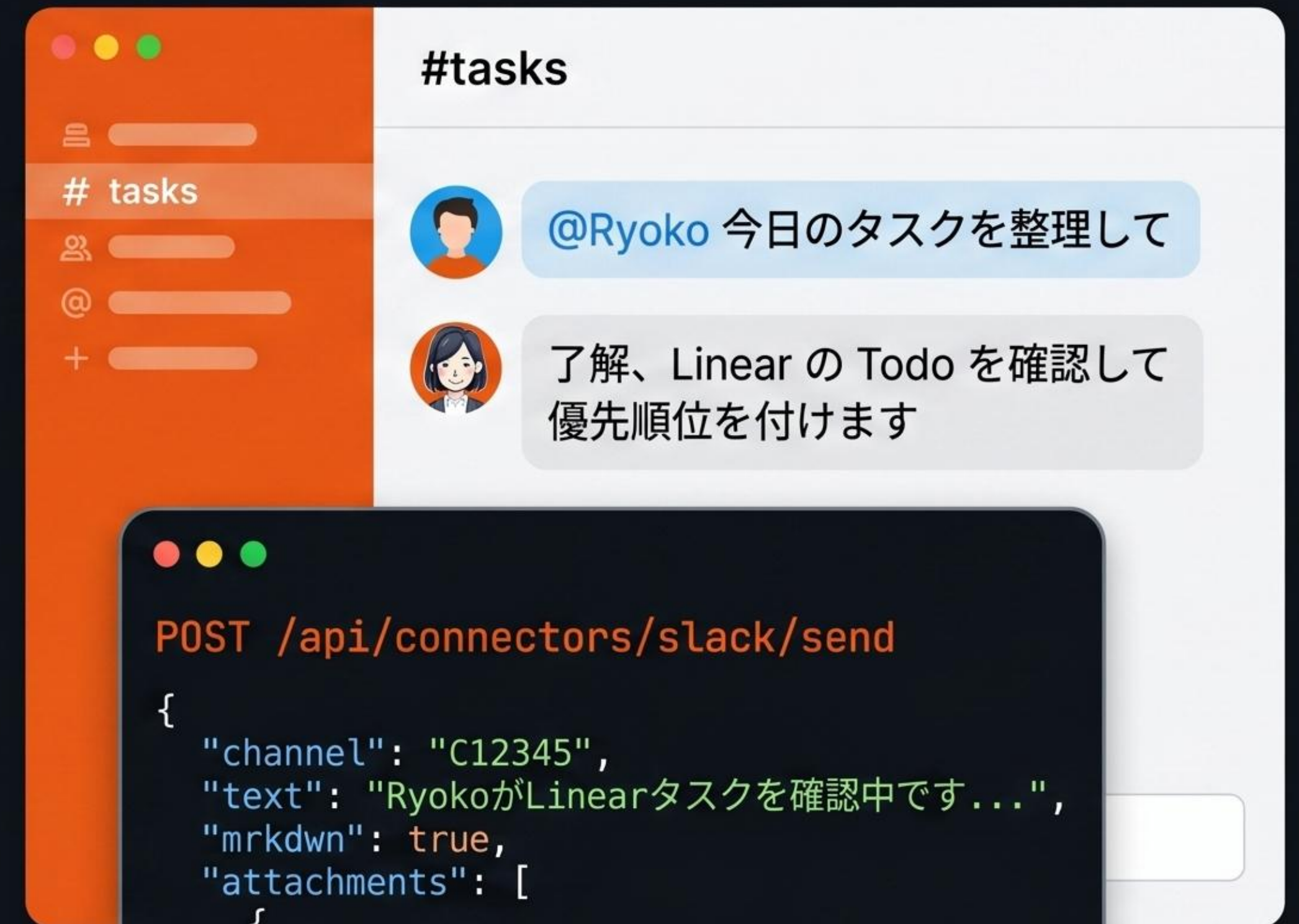
Slack ワークスペースに OpenRyoko ボットを統合し、ユーザーはチャンネルまたはダイレクトメッセージで **@Ryoko** をメンションして呼び出せます。



ユーザーの依頼に応じて、Ryoko は Linear などの外部タスク管理ツールと連携し、ToDo リストの確認、整理、優先順位付けを自動で行います。



開発者は、提供された REST API エンドポイント `POST /api/connectors/slack/send` を使用して、カスタムイベントに基づいた Slack メッセージ送信をシステムに実装可能です。



```
POST /api/connectors/slack/send
{
  "channel": "C12345",
  "text": "RyokoがLinearタスクを確認中です...",
  "mrkdwn": true,
  "attachments": [
    {
      "title": "タスクリスト確認中",
      "text": "優先順位を付けて整理します。"
    }
  ]
}
```

24pt



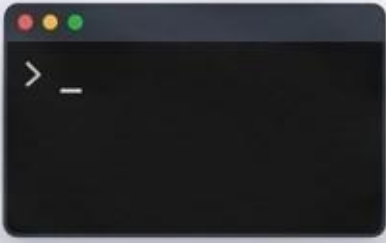





























› JSON 一行で、AI が定刻に動く

```
{  
  name: daily-briefing,  
  schedule: 0 7 * * *,  
  prompt: 今日の予定とニュースを要約して  
}
```

↻ ホットリロード対応

>競合比較_

他のエージェント基盤と何が違うか

	  OpenRyoko	 n8n	 LangChain	 Lindy
 OSS				
 Markdown 設定				
 永続記憶				
 Slack ネイティブ				
 cron				

\$ユースケース > _

「こんな使い方ができる」

01: 個人秘書



コーディング不要でMarkdownとYAMLを記述するだけで、Slack DMから予定管理、タスク作成、メールの草案作成・送信まで行う人格AI秘書を構築できる。

```
task:
  name: 振替 definition
  task: 個人秘書
```

```
$ npm i -g openryoko
&& ryoko start
```

稼働中

02: 営業チーム支援



Webサイトからの問い合わせへの自動一次返信や、オンライン会議ツールのAPIと連携し、議事録の自動取得・要約作成を常駐デーモンとして自動化する。

```
# markdown response
Response template:
  We ank you seene, and no tateno response2,
  ``response>
```

03: 研究PoC



cronを活用したタスク自動化や、自社データベースと連携させたエージェントアーキテクチャの実験基盤として、OpenRyokoを活用し迅速なプロトタイピングを行う。

```
$ npm execution
cron !* ** -> 087-0000

Agent:
  agent:
    name: "agent.agent"
    point: etc
```

ロードマップ

これからの計画

v2026.5



v2026.5 Multi-engine

コーディング経験ゼロの記者がClaude Codeを使い、Monday.comのクローンを1時間・コスト\$15で構築したのように、複数のAIエンジンを統合・強化する。

Q3



Q3 Multi-tenant

Q3にマルチテナント機能を実装したバージョンをリリースし、複数のチームが独立して利用可能にする。

Q4



SOC2 Enterprise

Q4までにSOC2コンプライアンスに対応した高度なセキュリティ機能を備えるエンタープライズ版を完了させる。

2027



2027 Marketplace

2027年に、AIのスキルや人格を安全に共有・売買できる『Marketplace』を一般公開する。

コミュニティ OSSで育てる、開かれた基盤



GitHub リポジトリ

Stars
☆ **1,234**

Forks
🔗 **567**

Issues
🔔 **89**

GitHubリポジトリでIssueの修正
やや新機能の提案を受け付けてい
ます。コードでの貢献（コントリ
ビューション）を歓迎します。

[[Contribute](#)]



Discord サーバー

Online Users
👤 **1,234**

Members
👥 **5,678**

公式Discordサーバーでは、技術
的な質問、ユースケースの共有、
ロードマップに関する議論が活発
に行われています。

[[Join Discord](#)]



Vibe Coder Bootcamp

Participants
👤 **500**

Events
📅 **10+**

初心者から熟練者までを対象とし
た、OpenRyokoを用いた自宅サー
バー構築とAI活用のためのハンズ
オンBootcampを開催予定です。

[[Learn More](#)]

5分で動く



```
$ npm i -g openryoko
```

依存インストール

```
$ ryoko init
```

config と人格生成

```
$ ryoko start
```

Gateway 起動

```
> _
```

● ● 稼働中

<http://localhost:7777> を開く



> **Star us on GitHub.** _

```
github.com/rsensui2/OpenRyoko  
$ npm i -g openryoko
```

#vibecoding

#aiagent

#oss