AIを使うときの画面切り替えが嫌で、ユーティリティツールを開発してたら、思ったより良くて、グローバルtoCプロダクトを開発している話

奥脇 真人　Masato Okuwaki

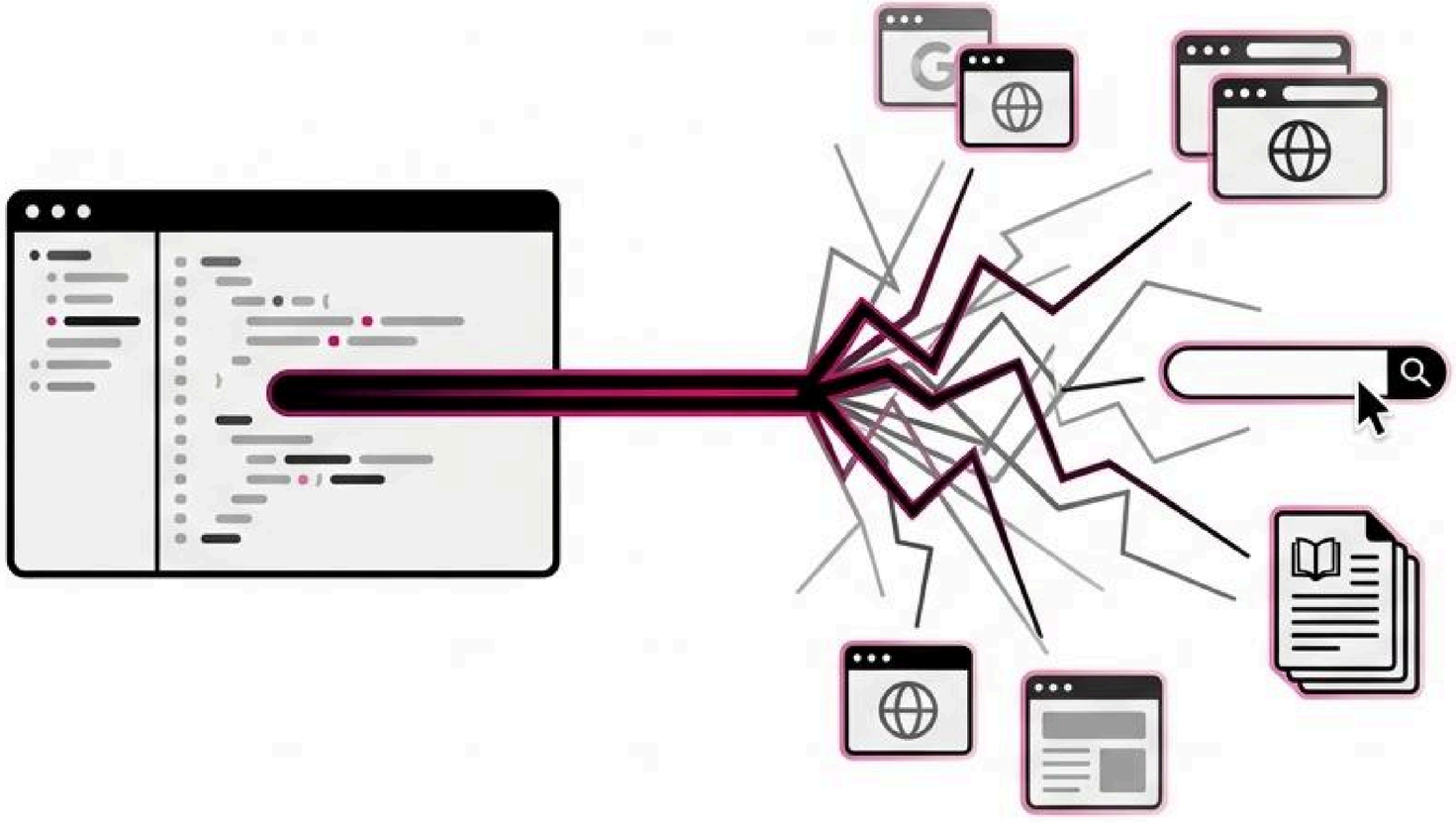
コーレ株式会社　代表取締役

AIプロダクト開発、AIコンサルティング

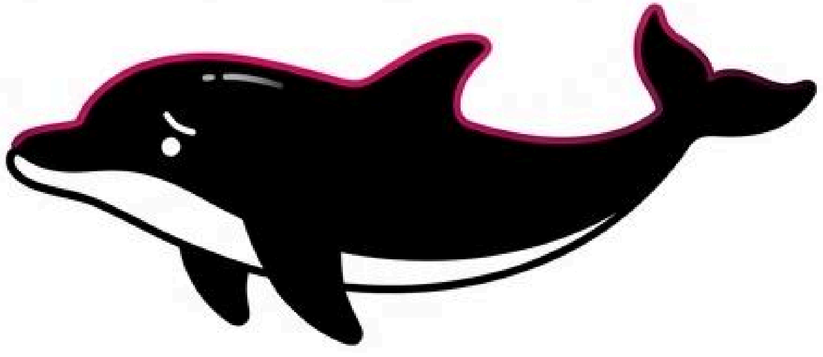
X：@okuwaki_m


AIツールを使うとき
画面切り替え多いの苦痛

その「ちょっと調べる」が、集中力を奪っていく。



複雑なコード、解読不能なエラー、
新しい技術文書…。
理解のためにブラウザを開いた瞬間、
私たちの集中は霧散します。
この「コンテキストスイッチ」の
コストは、想像以上に大きいのです。



自分のためだけの開発

macOS

画面常駐

高速

可愛いキャラ

プロンプト打たなくていい

図解も読める

画像もつくれる

動画もつくれる

ターミナル起動できる

思考を止めない、フルスクリーンAIアシスタント

画面上のあらゆるものを、ショートカット一つで瞬時に理解する。

テキストを選択:

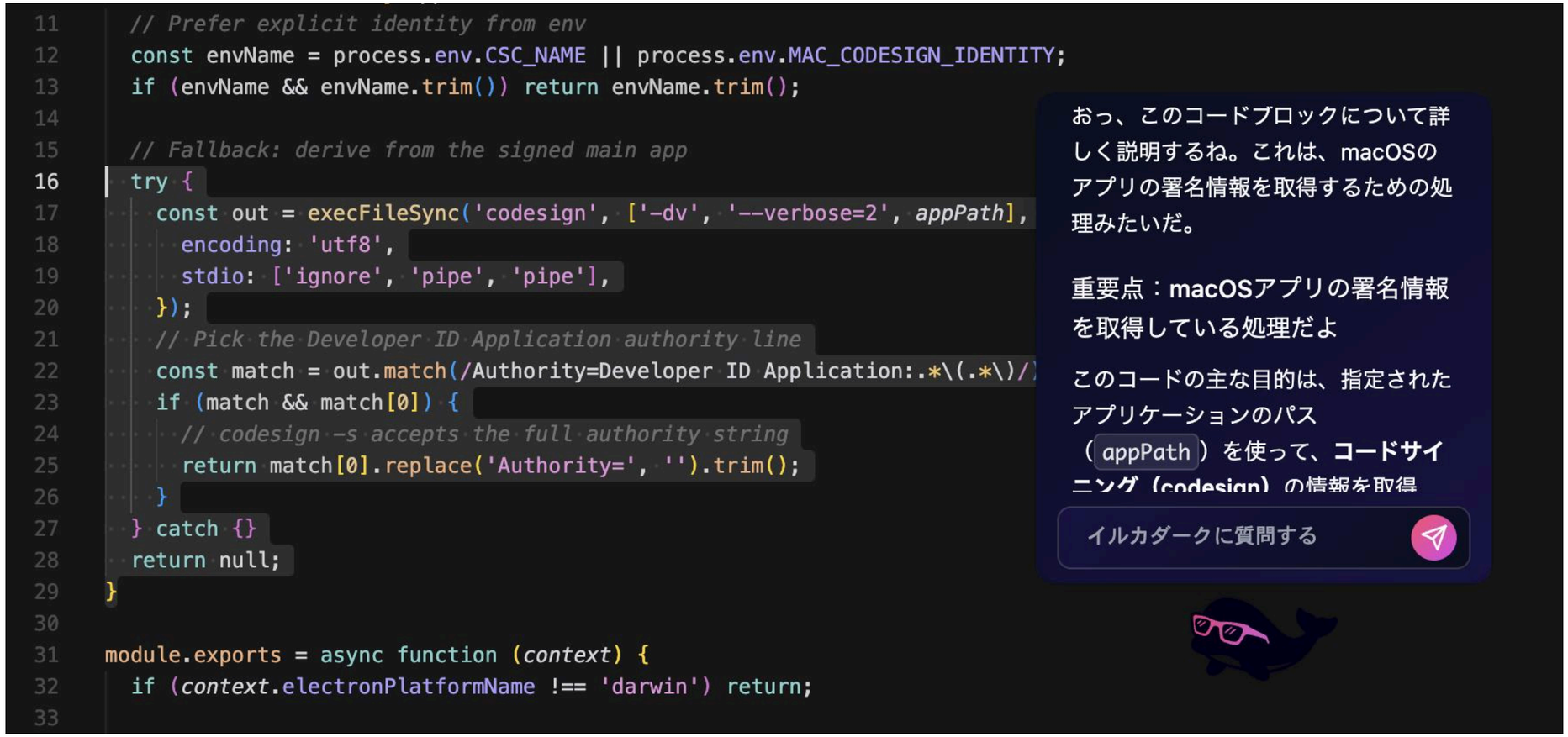
スクリーンショット:

URLを選択:

IrukaDark

画面を変えずに
ショートカットキーをポチッ

Cursorの上で
コードを選択してOption+A



Claude Codeの出力を選択して Option+A

AI技術論文のURLを選択して Option+Q

arXiv:2510.24690v1 [cs.AI] 28 Oct 2025

Bridging Tool Dependencies and Domain Knov
A Graph-Based Framework for In-Context Pla

Shengjie Liu

Amazon zycjlsj@amazon.com

Li Dong Amazon ldonga@amazon.com

Zhenyu Zhang Amazon zhenyuzh@amazon.com

Abstract

We present a framework for uncovering and exploiting dependencies among tools and documents to enhance exemplar artifact generation. Our method begins by constructing a tool knowledge graph from tool schemas—including descriptions, arguments, and output payloads—using a DeepResearch-inspired analysis. In parallel, we derive a complementary knowledge graph from internal documents and SOPs, which is then fused with the tool graph. To generate exemplar plans, we adopt a deep–sparse integration strategy that aligns structural tool dependencies with procedural knowledge. Experiments demonstrate that this unified framework effectively models tool interactions and improves plan generation, underscoring the benefits of linking tool graphs with domain knowledge graphs for tool-augmented reasoning and planning.

1 Introduction

よくわからない図解を選択して Option+S

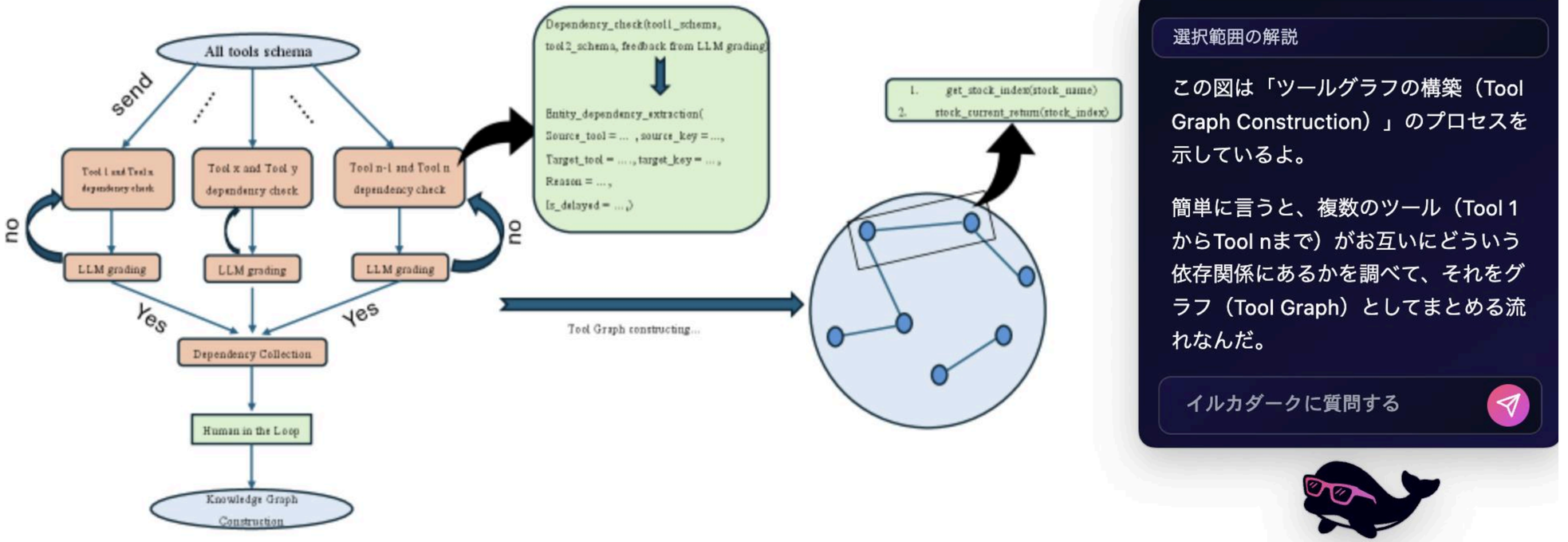


Figure 1: Tool Graph Construction

利用シーン①: Vibe Coding × エラーハンドリング

ノリと感覚でコーディングする人たち。
エラーが出たり、知らないツールを
使ったりする時も、画面を遷移せず
その場で解決したい。

「バイブスを途切れさせない」ために
使われている。

利用シーン②：熟練エンジニア × 未知の技術

AI駆動開発により、熟練者でも初めて触る言語や技術に取り組む機会が爆増。RAG、マルチモーダル等の最新論文や英語ドキュメントを必死に読む際に重宝されていた。

デスクトップツール？絶対にやりません。

Clipy, Alfred, Raycast...
市場は成熟した神ツールで溢れている。
ビジネスとして参入するのは正気の沙汰ではない。
レッドオーシャン。

「マーケットイン」は仕事でやる。

個人の開発では真逆を行く。「他人のニーズなんて知ったことか。
僕のパフォーマンスが上がればそれでいい」。

Market-In

Me-In

クライアントに見せたら「何それ！？」

自分専用のつもりで誰にも見せる予定はなかった。ある日、クライアント先のエンジニアの前でうっかり使ったら、驚かれた。「あれ、これ他人にも需要ある？」と初めて思った瞬間。

でもAIツールは日本では
説明工数が大変・・・

試しに海外向けにX投稿したら
1000件くらいダウンロードされた

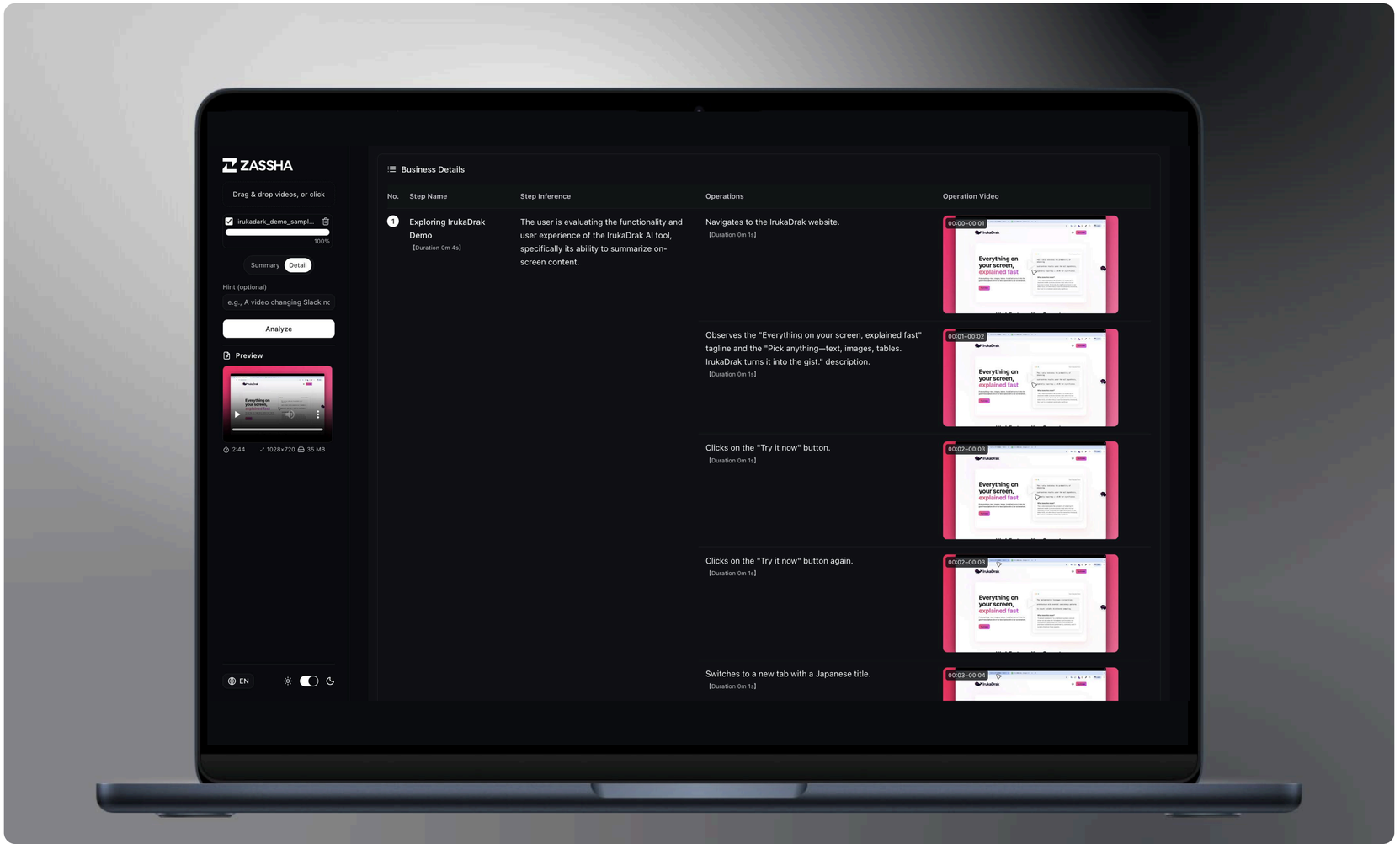
いけるかもしれない！

海外のtoCプロダクト展開を調査
プロダクトハントがいいらしい

とりあえず一回プロダクトハント出す用にプロダクトをつくった。結果10位...

ZASSHA

作業動画をアップロードすると、動画を分割して作業マニュアルをつくってくれるAI



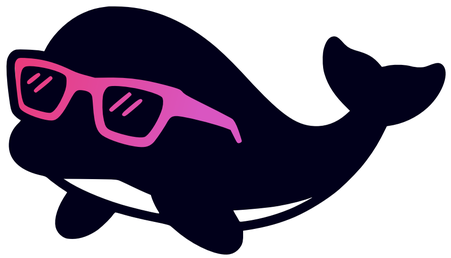
いろいろ聞いたり調査したら
しっかり準備するのが大切らしい

準備って…

AIで作れるものは増えた
しかし、広めるのが難しい
日本からグローバル攻めるのは
孤軍奮闘は難易度が高い

AI駆動開発で個人向けツールを高速でつくり
プロダクトハントで応援しあいましょう

X : @okuwaki_m



IrukaDark