

Boris Cherny ב-Odd Lots: איך Claude Code נבנה, ואיך מהנדסים עובדים עכשיו

ראש Claude Code ב-Anthropic מדבר על המודל כמנוע הצמיחה, על בטיחות כמוצר, על
מאות agents במקביל ועל ארגונים שעדיין מחזיקים מחשב אחד בפינה

Boris Cherny, בראיון ל-Joe Weisenthal ול-Bloomberg Odd Lots · Tracy Alloway (פוסט משותף ב-X מאת rari) ·
הועלה ל-X ב-2026-07-22

משך: כ-60 דקות · X

סיכום אוטומטי — AI Briefing (aibriefing.dev)

על המסמך הזה

זה פרק של הפודקאסט Odd Lots של Bloomberg. המקור מזוהה לפי הלוגו והאולפן בפריימים ולפי פתיחת הפרק בתמליל. המארחים Joe Weisenthal ו-Tracy Alloway מראיינים את Boris Cherny (ב-ASR הוא מופיע כ-"Churney"), שמוצג כיוצר וראש Claude Code ב-Anthropic. זו שיחה ולא הרצאה: אין שקפים, ויש רק הדגמה קצרה אחת של שרשור Slack לקראת הסוף.

לפי Cherny, הזינוק של Claude Code ב-2026 נובע כמעט כולו מהשיפור במודלים, ולא מה-harness. לדבריו בטיחות היא גם המשימה וגם יתרון מסחרי (prompt injection, sandbox, auto mode). עבודת המהנדס עלתה בשנתיים שתי רמות הפשטה: מכתובת קוד, לשיחה עם מודל, ומשם לשיחה עם מודל שמנהל מודלים אחרים. ארגונים ייהנו מזה רק אם יבנו את התהליכים מחדש סביב ה-AI, ולא ישאירו אותו בפינה.

איך לקרוא את המסמך

- **1. תקציר מנהלים:** הטענות המרכזיות עם חותמות זמן, ומה לא נאמר.
- **2. המקור והמנוע:** למה Anthropic בנתה מוצר קוד, dogfooding על ה-API הציבורי, ונקודות הצמיחה לפי גרסאות מודל.
- **3. בטיחות כמוצר:** prompt injection, תחרות ה-red team, sandbox, auto mode, והגלגול המבוקר של Mythos.
- **4. העבודה של המהנדס:** לולאת המשוב, שתי קפיצות הפשטה, אחוזי הקוד שכותב Claude, וחלוקת התפקידים החדשה.
- **5. אימוץ בארגונים:** סולם האימוץ, הגדרות ו-guardrails, פרטיות, הטענה בעד open source, חפירים (moats) ומיגרציות קוד.
- **6. הממשק:** מהטרמינל ל-Slack, הדגמת Claude שמצטרף לשרשור מעצמו, ומחקר HBR מ-1996.
- **7-8. מה עושים עכשיו ומילון:** פעולות מומלצות, שאלות פתוחות וטבלת מונחים.

הערת מקור המסמך הופק אוטומטית מהתמליל של ההקלטה ומהפריימים שלה. ה-ASR משבש שמות ומונחים: Claude מופיע לרוב כ-"Quad", ושמות כמו Fable, Mythos ו-"tag" מסומנים כלא ודאיים במקומות שבהם הם מופיעים. הציטוטים מועתקים מילה במילה מהתמליל, כולל השיבושים, ונבדקו מול חותמת הזמן שלהם. לא נוספו עובדות ממקורות חיצוניים. התמליל נקטע באמצע משפט סמוך לדקה 59.

1. תקציר מנהלים

השיחה פונה לקהל עסקי ולא למפתחים. לכן Cherny מדבר בה מעט על ארכיטקטורה ועל workflow, והרבה על בטיחות, על ארגונים ועל השינוי בתפקיד המהנדס. רוב הטענות נשענות על ניסיון פנימי של Anthropic ועל אנקדוטות מלקוחות, ולא על נתונים מתוקפים.

- **הצמיחה מגיעה מהמודל:** לשאלה אם ה-harness או המודל הציתו את הגל הוא עונה "Oh, it's almost all the model" ומונה נקודות צמיחה ב-Opus 4 (מאי), Opus 4.5 (נובמבר), Opus 4.6 (פברואר) ועכשיו [9:23] Fable.
 - **אין API סודי:** Claude Code רץ על אותו מודל ציבורי ועל אותו Anthropic API שהלקוחות משתמשים בו. לכן שיפור במודל מגיע גם ללקוחות [9:23].
 - **Prompt injection לא עבר:** בתחרות של שבוע עם פרס של 20 אלף דולר, חוקרים חיצוניים הצליחו להזריק prompt לכל המודלים חוץ מ-Claude בתוך Claude Code. הוא מייחס את זה ל-alignment, ל-probes של interpretability ול-auto mode [12:27].
 - **הקוד של המודל טוב יותר משלו:** לדבריו, מאז Opus 4.7 או 4.8 ובוודאות מאז Fable, הקוד שהמודל כותב כמעט תמיד טוב יותר מהקוד שהוא היה כותב בעצמו [20:04].
 - **לולאת משוב חשובה כמו האיכות במעבר הראשון:** משל הפסל עם כיסוי העיניים: בלי דפדפן, simulator או הרצה end-to-end, התוצאה לא תגיע לפוטנציאל שלה [21:35].
 - **שתי קפיצות הפשטה בשנתיים:** אחרי 50 שנה של כתיבת תוכנה ישירה, היום מדברים עם מודל שמדבר עם מודלים אחרים. בכל רגע רצים אצלו כמה מופעים של "sometimes hundreds, sometimes thousands" Claude, [24:39].
 - **100% מהקוד שלו:** מאז נובמבר כל הקוד שלו נכתב ב-Claude Code. זה נכון גם לכל Claude Code ו-Cowork, והמוצק ב-Anthropic הוא "something like 90%" [26:12].
 - **אימוץ בארגונים הוא סולם:** מתחילים מ-Claude אחד למהנדס ועוברים ל-10, ל-100 ולפעמים ל-1,000. לפי הדוגמאות שלו, ארגונים שעושים את זה הם בנקים גדולים בניו יורק, חברות פארמה ו-[40:07] NASA [38:34].
 - **מיגרציה כבר כלכלית:** מהנדס אחד בצוות Bun העביר את כל בסיס הקוד מ-Zig ל-Rust בכ-11 ימים. בשיחה הוזכר מחיר של כ-150 אלף דולר בקרדיטים, ומאותו היגיון זה תקף גם ל-COBOL בבנקים [50:54].
- מה לא נאמר**
- **אין פירוט טכני של עבודה עם מאות agents:** הוא לא מסביר איך מתאמים אלפי מופעים, איך מונעים התנגשויות, מה עולה ה-compute ומי עושה review לקוד.
 - **סתירה באחוזים:** Cherny אומר "something like 90%", ומיד אחר כך המארח שואל על ה-"2%" שנשאר. המספר המדויק לא הובהר [26:12].
 - **תחרות ה-prompt injection:** לא נמסר כמה חוקרים השתתפו, אילו מודלים מתחרים נבדקו ומה המתודולוגיה. יש רק הפניה ל-model cards של Opus 4.8 ו-Sonnet 5.
 - **העלות של מיגרציית Bun:** הסכום של 150 אלף דולר נאמר בשיחה ולא ברור מהתמליל מי אמר אותו. Cherny לא אישר ולא תיקן.
 - **אין מספרים עסקיים:** אין הכנסות, אין מספר משתמשים ואין מדד צמיחה מעבר ל-"growth inflected".
 - **פרטיות:** הטענה שאפשר להוכיח ש-Anthropic לא רואה שיחות לא פורטה ברמת מנגנון.

2. המקור והמנוע: למה מוצר קוד, ולמה המודל

לפי Cherny, Claude Code נולד מתוך סדר העדיפויות של הבטיחות. evals ו-mechanistic interpretability לא מספיקים, ובשלב מסוים צריך לראות איך אנשים משתמשים במודל בעבודה אמיתית. מודלים פועלים בעולם דרך קוד, "They don't have bodies like we do", ולכן קוד היה הזירה הטבעית [4:44]. הצוות שבו התחיל Anthropic Labs, בנה לדבריו את Claude Code, את MCP, את Skills ואת אפליקציית הדסקטופ [0:00]. נקודת הפתיחה הטכנית הייתה Sonnet 3.5, שלדבריו היה מודל הקוד הטוב הראשון ואיפשר לעבור מהשלמה של שורה לכתיבה של קובץ שלם ואפילו פיצ'ר שלם [6:18].

לגבי הצמיחה הוא חד: "Oh, it's almost all the model" [9:23]. כל גרסת מודל גדולה הביאה נקודת צמיחה ב-Claude Code. הוא מדגיש שזה בכוונה, כי Claude Code משתמש באותו API ציבורי כמו הלקוחות: "We don't

[9:23] "have some secret API that we use". כשהמודל משתפר, גם הלקוחות שבונים על הפלטפורמה רואים את אותה צמיחה [10:55].

Boris Cherny [0:00] I remember we were building early prototypes of the desktop app, and that — had the first ever versions of computer use when we were first starting to crack it. And we asked Quad to, I think it was like we asked it to order a pizza

האנקדוטה ממשיכה: אחרי שהזמין פיצה, המודל "השתעמם" ונכנס לקרוא את Hacker News. Cherny משתמש בזה כדי להמחיש את ההבדל מהיום. לדבריו, השילוב של alignment, אינטליגנציה כללית וזיכרון הוא שמאפשר session שרץ שבועות: "[0:00] Like I have tag sessions that have been running for weeks at a time". מהתמליל לא ברור מה זה "tag". סביר שזה שם של מוצר Claude שעובד ב-Slack, אבל השם לא ודאי.

3. בטיחות כמוצר: auto-prompt injection, sandbox mode

לדבריו, CISOs חוששים בעיקר מ-prompt injection: דף אינטרנט שהמודל קורא ובו הוראה זדונית כמו "delete all the files" [10:55]. ההגנה המקורית ב-Claude Code הייתה בקשת האישור הידועה (yes/no) לפני פקודה מסוכנת [12:27]. היום, לדבריו, ההגנה נשענת על שלוש שכבות: alignment שנמצא בתוך המודל, probes של interpretability שמזהים ב-"נירונים" של המודל שמנסים להזריק לו prompt, ו-auto mode, מצב הרשאות חדש שבו אין יותר בקשות אישור והוא גם בטוח יותר [12:27]–[13:57].

Boris Cherny [12:27] There's a bunch of researchers that participated. There's a bunch of other — models in the mix. They were able to prompt inject every single model, except for our model in quad .code

לגבי מודל שעוקף מגבלות כדי להשיג מטרה, הוא מתאר שני מנגנונים. הראשון הוא alignment, כלומר אימון המודל לא "לפרוץ" אם המשתמש לא רוצה בזה. להשתמש בתשתית חלופית שעובדת זה בסדר, לפרוץ מערכת זה לא. השני הוא guardrails. ב-Anthropic מריצים את Claude Code בתוך sandbox שמגביל גישה רק לקבצים ולאתרים שהמשתמש אישר. ה-sandbox הוא open source ועובד עם כל agent. לשאלה אם המודל פורץ ממנו התשובה היא "It can", ולכן מבצעים red teaming ו-penetration testing באופן שוטף [17:00].

על Mythos (השם מה-ASR, לא ודאי): לפי Cherny הוא מודל מיוחד שטוב מאוד במציאת zero-day, ולכן נמסר קודם "לטובים" במסגרת Mythos access on day one, everyone would just kind of be hacking. [33:59] לדבריו, Fable הוא הגרסה של Mythos בלי יכולות הפריצה האלה, והיא זמינה לכולם. הוא עצמו עובד איתה כל יום [35:31].

4. העבודה של המהנדס: לולאות, הפשטה ותפקידים

למה המודל טועה בפעם הראשונה ומתקן אחר כך? Cherny משווה את זה לטייטה ראשונה של טקסט: קוד הוא פעולה יצירתית, ולא תהליך שמגיע ישר לתשובה [18:32]. הטענה החשובה יותר שלו היא שאיכות המעבר הראשון היא רק חצי מהתמונה. החצי השני הוא לולאת המשוב: "if Quad can't test the website it's building in a browser, if it can't open the iOS app it's building in an iOS simulator ... it's just not going to be as good as it could have been" [21:35].

Boris Cherny [24:39] So you talk to the model, the model talks to other models. Those models — write the source code. And this is crazy because we've been stuck in this one place for 50 years, and we just had two leaps in two years

הוא ממקם את זה בהיסטוריה: מכרטיסים מנוקבים (סבא שלו תכנת כך בברית המועצות), דרך לוגיקה שנבנתה בשבבים ב-Apple I, ועד מערכות הפעלה בתוכנה. היום, לדבריו, הוא כבר לא עובד ימים או שבועות על תוכנה אחת. בכל רגע רצים אצלו כמה מופעים של Claude, לפעמים מאות ולפעמים אלפים, "and the funny thing is, I just ... never run out of things for them to do" [24:39].

המספרים: "100% [26:12] of my code has been written by quad code since November of last year". כל Claude Code ו-Cowork נכתבים כך, וגם אחוז הולך וגדל מקוד התשתית והמחקר. מה שנשאר ידני הוא בעיקר שינוי של שני תווים בקובץ קונפיגורציה. בהרצאות ב-Y Combinator, שיעור האנשים שכותבים 100% מהקוד עם Claude Code עלה מרבע לקצת יותר מחצי [27:43].

מה עושה (לפי Cherny)	תפקיד
מוצאים את הרעיון הראשון ומבצעים איטרציות מהירות	Prototypers
לוקחים רעיון חדש ומביאים אותו לשוק כמוצר	Builders
מתחזקים תוכנה שכבר פועלת בקנה מידה	Maintainers
מגדילים מוצר עם product-market fit פי 10 ופי 100, ולדבריו מבוקשים מאוד ב-Anthropic	Growers / Scalars
מלטשים את המוצר, את התשתית ואת הקוד ומסירים קצוות חדים. השם "Perfactors" הוצע על ידי המארחים	Sweepers ("Perfactors")

הרקע לחלוקה: בצוות Claude Code כולם כותבים קוד, כולל מעצבים, מנהלי מוצר ומנהלי הנדסה. המעצבת לא צריכה לבקש ממנו להזיז כפתור בפיקסל [29:20]. לגבי שפות תכנות הוא אומר "I think they're largely irrelevant" [50:54]. טיפוסים ו-static analysis עוזרים למודל, אבל פחות ופחות ככל שהוא משתפר. הוא צופה "Cambrian explosion" של שפות חדשות ולא סטנדרטיזציה [52:30].

5. אימוץ בארגונים: סולם, הגדרות, פרטיות ו-moats

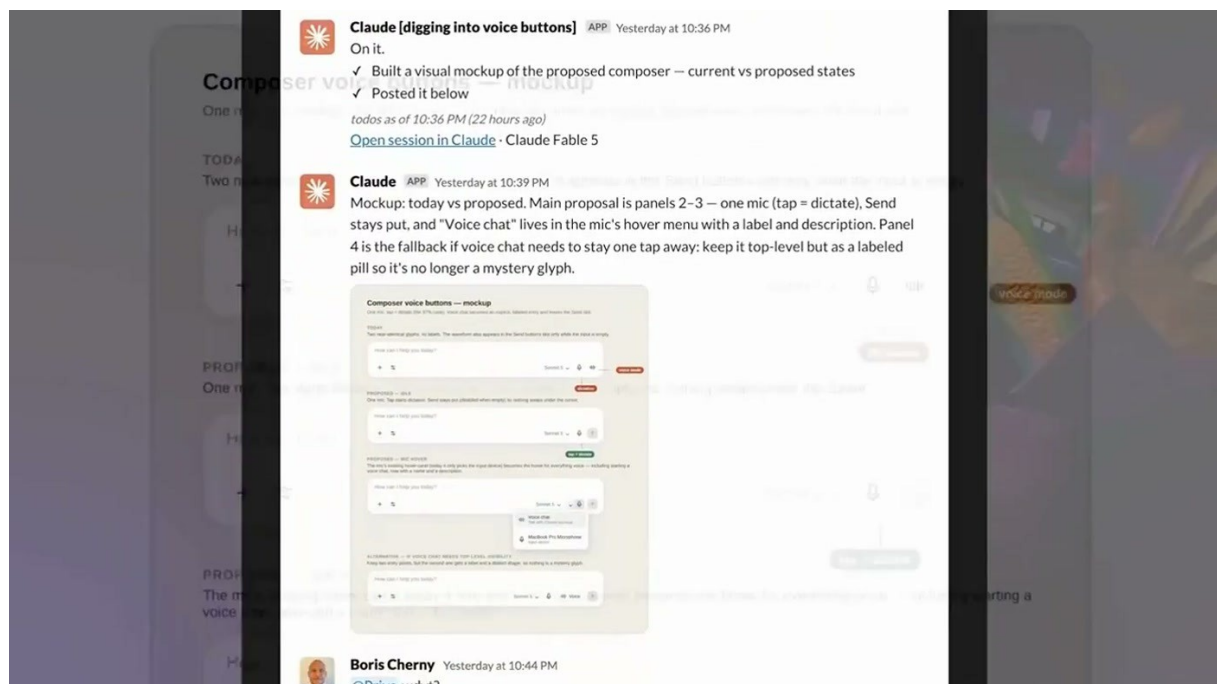
Cherny מתאר אימוץ בארגון כסולם [37:03]. השלב הראשון הוא AI בתוך IDE. בשלב השני כל עובד מקבל Claude Code או Cowork, בהתחלה ביחס של session אחד לעובד. משם עולים ל-10, ל-100 ולפעמים ל-1,000 מופעים של Claude למהנדס [38:34]. ה-guardrails שהוא מזכיר הם בקרת הוצאות לכל משתמש, advisor models, בחירת רמת effort ברמת הארגון ו-sandboxing, עם ברירות מחדל בטוחות. יש לדבריו "probably four or five hundred" הגדרות, ואפשר לבקש מ-Claude להגדיר אותן, כי "quad knows its own settings" [40:07]. ארגונים מעדיפים בדרך כלל תשלום לפי טוקן ולא מנוי, כדי לשלוט בעלויות ולא להיתקע ב-rate limits [35:31].

המארחים מעלים את הטענה שמיוחסת ל-Microsoft ול-Alex Karp, שלפיה מי שמחבר את Anthropic לעסק "מכניס שועל ללול". Cherny מציע קודם לבדוק את התמריצים של מי שאומר את זה. אחר כך הוא עונה בשני חלקים [46:16]–[49:21]. החלק הראשון הוא פרטיות: כשמשתמש מדווח על באג, "I cannot see that data", ואפשר להוכיח שאיש ב-Anthropic לא יכול לראות את השיחה. החלק השני כבד יותר לדבריו: המודלים ממשיכים להשתפר. אם האינטליגנציה הייתה קופאת, אולי היה טעם לשלוט בתשתית בעצמך. כל עוד ה-frontier זז, אירוח עצמי פירושו לוותר עליו ולשאת בעצמך בעלות התפעול של inference.

על ה-"SaaSocalypse" הוא עונה דרך מסגרת Claude Code 7. Powers שוחק חפירים כמו switching costs, כי אפשר לבקש ממנו "port me". אבל חברות גדולות נשענות על שילוב של כמה חפירים, למשל network effects או cornered resource, ורובם לדבריו נשארים חזקים כמו קודם [43:11]–[44:46]. בנושא COBOL הוא מביא את Bun: Jared מצוות Bun העביר את בסיס הקוד מ-Zig ל-Rust בכ-11 ימים, לבד, עם Claude Code ועם משהו שה-ASR כתב כ-"dynamic core close" (המונח לא ודאי). בעבר זה היה לוקח לכמה מהנדסים כשנה, והם לא היו עושים את זה בכלל [50:54].

6. הממשק: מהטרמינל ל-Slack

לשאלה אם ה-CLI היה פתרון זמני, Cherny עונה ש-Claude Code יצא מהטרמינל מהר: יש extensions ל-IDE, אפליקציית דסקטופ שמאחדת chat ו-cowork, ואפליקציות ל-iOS ול-Android. היום הוא עובד בעיקר דרך Slack, ולפני כן עבד בעיקר מהטלפון. בטרמינל הוא משתמש רק לפעמים [30:52]–[32:24]. כדי להראות איך נראה ממשק גרפי בעיניו, הוא מציג שרשור מערך feedback פנימי.



הדגמה מ-Slack: שרשרת על שני כפתורי האודיו שמבלבלים משתמשים. Claude מדווח שבנה mockup ויזואלי של ה-composer המוצע לעומת הקיים, ומצרך פאנלים להשוואה

מה שקרה בשרשרת, לפי התיאור שלו [54:07]–[55:37]: הוא פרסם תלונה על שני אייקוני אודיו שמבלבלים אותו. Claude הצטרף לשרשרת בלי שתייג אותו ("It just noticed this thread and it jumped in and it responded"). הוא שלף נתוני שימוש בכל כפתור משני מקורות, Google BigQuery ו-Datadog, הציע חלופות, ולפי בקשה צייר mockups. אחר כך תויגה המעצבת והשיחה הפכה ל-multiplayer. בפריים הסמוך מופיעים גם שמות של משתתפים נוספים ותגובות שלהם להצעה. לדבריו, "This is actually how we write most of our code now at Anthropic" [55:37].

לשאלה על חיכוך בארגונים גדולים, למשל עובד שזה היה התפקיד שלו, Cherny מביא מאמר ב-Harvard Business Review מ-1996 (לפי זיכרונו) שבדק למה המחשב האישי לא שיפר את הפריון. חברות ששמו מחשב בפינה ומינו מישהו להקליד לתוכו לא הרוויחו כלום. החברות שהרוויחו שמו את המחשב במרכז, עשו דיגיטציה לכל התהליכים וזרקו את ארונות התיקים. אחר כך מצאו את צוואר הבקבוק הבא ועשו לו דיגיטציה, וכך הלאה [57:11]–[58:43].

Boris Cherny [58:43] they are not seeing productivity benefits instead it's just someone's job to — talk to the computer

7. מה עושים עם זה

- **תנו ל-agent לראות את התוצאה:** לפני שמשפרים prompts, ודאו שה-agent יכול להריץ דפדפן, simulator או שירות end-to-end ולבדוק את העבודה שלו. זו הטענה הכי מעשית בשיחה [21:35].
- **טפסו בסולם בהדרגה:** מ-session אחד לעובד, עם בקרת הוצאות וברירות מחדל בטוחות, ורק אחר כך הרצה מקבילית רחבה.
- **הוסיפו sandbox לכל agent:** לפי Cherny ה-sandbox של Anthropic הוא open source ועובד עם כל agent. כדאי לבדוק את זה לפני שמוותרים על בקשות האישור.
- **בחנו מחדש מיגרציות שנדחו:** אם מעבר בין שפות לקח שבועות ולא שנה, כדאי לחשב מחדש את העלות של פרויקטי legacy כמו COBOL או מעבר שפה.
- **אל תשאירו את ה-AI בפינה:** לפי מחקר HBR, מזהים צוואר בקבוק בתהליך, בונים אותו מחדש סביב ה-agent, ועוברים לצוואר הבקבוק הבא.
- **הגדירו מחדש תפקידים בצוות:** שווה לבדוק אם החלוקה sweeper, prototyper, builder, maintainer, scaler, PM, engineer, ו-designer מתארת את הצוות שלכם טוב יותר מהחלוקה.

שאלה פתוחה	למה היא חשובה
איך מתאמים מאות או אלפי מופעים של Claude בפועל?	Cherny לא תיאר את ה-orchestration, את ה-review או את העלות. זה הפער הגדול בין הטענה ליישום
מה האחוז האמיתי של קוד שנכתב על ידי AI ב-Anthropic?	בתמליל נאמר "90%" ואז הוזכרו "2%" שנשארו. המספר לא הובהר
מה המתודולוגיה של תחרות ה-prompt injection?	זו הטענה הבטיחותית החזקה בשיחה, אבל אין בה פרטים מעבר להפניה ל-model cards
האם auto mode בטוח יותר מבקשות אישור גם מחוץ ל-Anthropic?	הטענה נאמרה בלי נתונים
האם גלגול מבוקר של מודלים מסוכנים יהפוך לנורמה?	Cherny הציג את Mythos כמקרה מיוחד, אבל לא התחייב לגבי מודלים עתידיים

8. מילון מונחים

מונח	משמעות
Harness	שכבת התוכנה שעוטפת את המודל, כאן Claude Code, ונותנת לו כלים, הרשאות וממשק
Dogfooding	שימוש פנימי במוצר של החברה עצמה, כאן דרך אותו API ציבורי
Prompt injection	הוראה זדונית שמוסתר בתוכן שהמודל קורא ומנסה להשתלט על הפעולות שלו
Mechanistic interpretability	מחקר על מה שקורה בתוך הרשת. כאן משמש לבניית probes שמזהים הזרקה
Auto mode	מצב הרשאות ב-Claude Code שבו אין בקשות אישור, ולפי Cherny הוא בטוח יותר
Sandbox	גבול שמגביל את ה-agent לקבצים ולאתרים מאושרים. לפי Cherny הוא open source
Red teaming	ניסיון מכוון לפרוץ או להטעות את המערכת כדי למצוא חולשות
Alignment	אימון המודל לעשות מה שהמשתמש התכוון ומה שטוב לאנשים באופן כללי
Scaling laws	הקשר בין compute, data וגודל הרשת לבין אינטליגנציית המודל
Zero-day	חולשת אבטחה שעדיין לא ידועה ליצרן ולכן אין לה תיקון
Moat / 7 Powers	חפיר עסקי. מסגרת של שבעה מקורות של יתרון מתמשך, כמו scale, network effects ו-switching costs
Rate limits	מגבלות שימוש במנויים. ארגונים נמנעים מהן בתשלום לפי טיקן
Product-market fit	התאמה בין מוצר לביקוש בשוק, שלב שבו עובר המוצר מ-builders ל-scalers