

בנק בנה coding agent משלו, והוא המריא: Agent Chip של Monzo

איך תשתית פנימית קיימת (MCP gateway, sandbox, אינטגרציות) הפכה וemoji
ב-Slack לסוכן שכותב יותר מ-10% מה-PRs

2026 באוקטובר Felipe Felisola Caso, Francesco Galletta · AI Engineer · 20

משך: כ-16 דקות · YouTube

סיכום אוטומטי — AI Briefing (aibriefing.dev)

על המסמך הזה

זו הרצאה קצרה, כ-16 דקות, מכנס AI Engineer בניו יורק (לפי השקופיות: 13 באוקטובר 2026, McCarthy Room). מרצים בה שני מהנדסים מ-Monzo, הבנק הדיגיטלי הבריטי: Francesco ו-Felippe Felisola Caso. הם מספרים על Agent Chip, ה-coding agent הפנימי של Monzo. הוא התחיל כ-emoji שמגיבים איתו על הודעה ב-Slack, והיום הוא כותב יותר מ-10% מה-PRs שממוזגים בכל ארגון הטכנולוגיה ומריץ יותר מ-1,800 משימות ביום.

הטענה המרכזית שלהם היא שהסוד לא נמצא במודל אלא בפלטפורמה. בהרצאה, LLMs זה קיצור של Lots of Little Multipliers: הרבה יכולות פלטפורמה שכבר היו קיימות (alerting, monitoring, staging, תורים, deployment), ועליהן MCP gateway ומערכת agent orchestration שמריצה סוכנים ב-sandbox. ברגע שהסוכן עצמו הפך לאבן בניין, נבנו מעליו בקלות שימושים חדשים: ניהול incidents, on-call לאיכות נתונים, ערוצי שאלות בין צוותים ושינויים שמגיעים מאנשים שאינם מהנדסים.

איך לקרוא את המסמך

- 1. **תקציר מנהלים:** הטענות המרכזיות והראיות, ובסוף מה לא נאמר.
- 2. **ההקשר:** גודל הבעיה ב-Monzo ואיך Agent Chip הפך ויראלי.
- 3. **Lots of Little Multipliers:** למה הפלטפורמה הקיימת היא הסיבה להצלחה.
- 4. **MCP gateway ו-sandbox:** הארכיטקטורה, כלומר מה חשוף לסוכן, איך הוא מבודד ואיך ה-egress נשלט.
- 5. **הסוכן כאבן בניין:** incident mode, ערוצי ask, on-call לנתונים ותרומות של אנשים שאינם מהנדסים.
- 6. **למה להמשיך in-house: composability,** אי-תלות בספק והרחבת חלון הפיתוח.
- 7. **מה לעשות עם זה:** צעדים מעשיים וטבלת שאלות פתוחות.
- 8. **מילון מונחים:** המונחים באנגלית שמופיעים במסמך.

הערות מקור המסמך הופק אוטומטית מהתמלול ומהשקופיות של ההקלטה. ה-ASR משבש שמות: שם הסוכן מופיע בתמלול כ-Aging Chip, Agentship ו-agent chip, ובשקופיות כ-agent-chip#, ולכן המסמך משתמש ב-Agent Chip. מה שהתמלול כותב CloudTag מופיע בשקופיות כ-Claude Tag. מי אמר מה הוסק מההקשר (הדובר השני מפנה ל"מה ש-Francesco הזכיר"), ולכן הייחוס לא ודאי. הציטוטים מועתקים מילה במילה מהתמלול ונבדקו מול חותמת הזמן שלהם. לא נוסף מידע ממקורות חיצוניים.

1. תקציר מנהלים

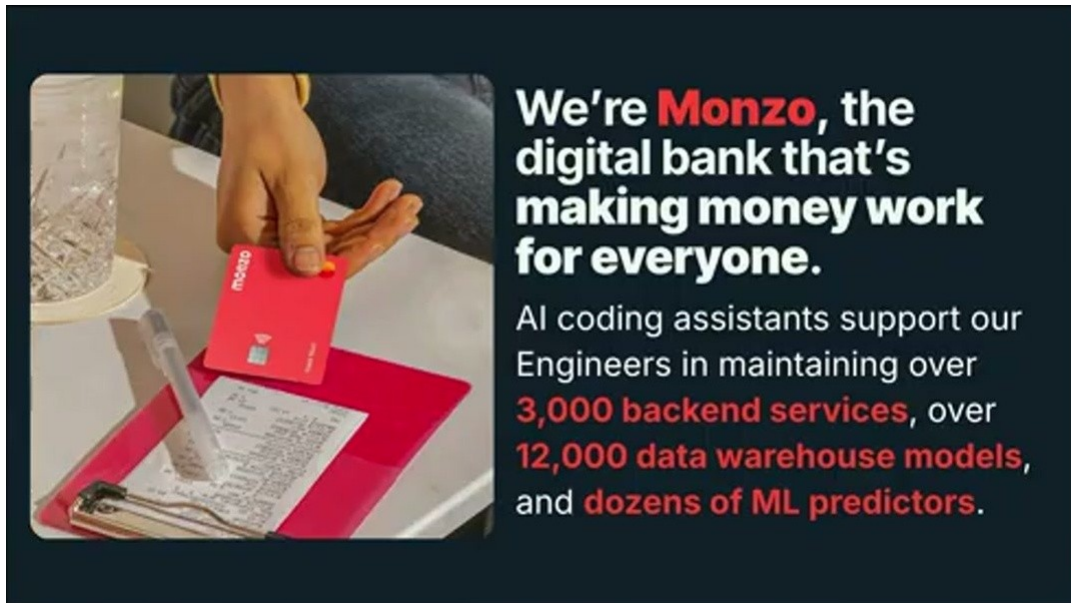
זו לא הרצאה על מודל או על prompt engineering. זו הרצאה על Monzo platform engineering. הצליחה לבנות coding agent פנימי מהר, ושהוא מייצר ערך, כי רוב החלקים הקשים כבר היו קיימים אצלה. היא גם טוענת שהסוכן עצמו הפך לרכיב פלטפורמה שאפשר לבנות עליו שימושים נוספים.

- **אימוץ בהיקף משמעותי:** לפי השקופית, Agent Chip כותב יותר מ-10% מה-PRs הממוזגים בכל ארגון הטכנולוגיה (data, ML, backend, app) ומריץ יותר מ-1,800 משימות ביום.
- **נקודת כניסה כמעט בלי חיכוך:** מגיבים על הודעה ב-Slack עם emoji, וההודעה הופכת ל-prompt. הערוץ הפך ויראלי בינואר 2026.
- **הסוד הוא הפלטפורמה ולא המודל:** Lots of Little Multipliers, כלומר alerting, monitoring, אינטגרציות, staging, תורים, deployment ושכבת database שכבר היו קיימים.
- **MCP gateway קדם לסוכן המרוחק:** הוא נבנה קודם עבור סוכנים מקומיים. הוא חושף בצורה דטרמיניסטית רק את הכלים ומקורות הידע שנבחרו, ומנהל authentication ו-authorization.
- **בידוד מוחלט מה-production:** הסוכנים רצים ב-pods בתוך namespace מבודד. ה-image מגדיר אילו כלים זמינים, ו-proxy מאפשר רק egress שהוגדר במפורש. ההצדקה: "we are a bank".
- **הסוכן כאבן בניין:** Slack, GitHub ו-Linear מחוברים אליו, ומעליו נבנו incident mode, ערוצי squad-ask ו-Agent Chip On Call לאיכות נתונים, בגרסה תגובתית ובגרסה יזומה.
- **אנשים שאינם מהנדסים מגישים טיוטות:** הם מקבלים טיוטה שכבר נבדקה ב-staging, ורק אחר כך מהנדס נכנס ל-review ול-deploy.
- **אי-תלות בספק:** את קופסת הסוכן אפשר להחליף בכל מודל, ולכן Monzo לא קשורה לספק מסוים.

מה לא נאמר

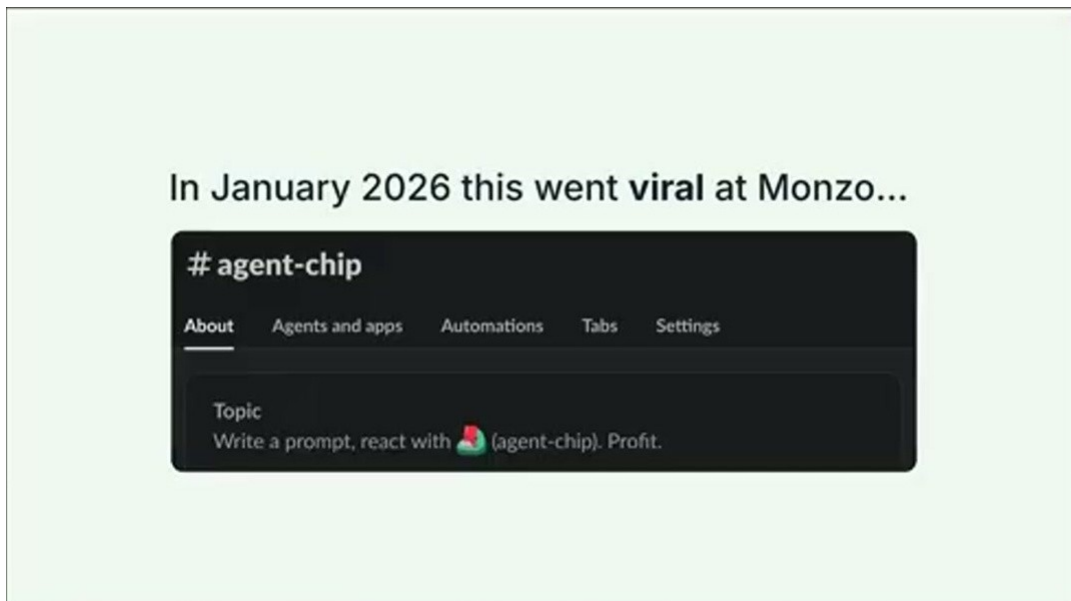
- **אין מדדי איכות:** לא נאמר מה אחוז ה-PRs של הסוכן שעוברים review בלי תיקונים, מה שיעור ה-revert או ה-bugs, ואיך בדיוק נמדד ה-10%.
- **אין עלויות:** לא נמסרו עלות למשימה, עלות חודשית או השוואה למוצרים מסחריים.
- **לא נאמר אילו מודלים:** נאמר רק שהפלטפורמה provider-agnostic. לא נאמר באיזה מודל או harness משתמשים בפועל.
- **אין פרטי מימוש:** טכנולוגיית ה-sandbox, מספר המהנדסים בצוות ומשך הבנייה לא פורטו. המונחים pods ו-namespace מרמזים על Kubernetes, אבל זה לא נאמר במפורש.
- **אין סיפורי כשל:** לא סופר על תקלות, פריצות או החלטות שהיו עושים אחרת.

2. ההקשר: בנק, 3,000 שירותים ו-emoji ב-Slack



Monzo במספרים: יותר מ-3,000 backend services, יותר מ-12,000 data warehouse models ועשרות ML predictors

כדי להמחיש את גודל הבעיה, הדובר מתאר צוותים שמתחזקים יותר מ-3,000 שירותי backend וכ-13,000 מודלים ב-data warehouse (בשקופית כתוב 12,000), ועוד עשרות predictors שמחליטים על אשראי ועל financial crime. הפתיחה מצטטת ציוץ של Gergely Orosz מכתבה על מצב התעשייה ב-2026: "are you even a serious tech company?". הוא מוצג כבדיחה, אבל גם כרקע.



הערוץ #agent-chip ב-Slack, עם ה-Profit, [emoji], react with prompt, Write a topic:

— Francesco Galletta (ייוס משוער) You can give it any prompt directly by just reacting with the little emoji to the message, and it will pick it up as a prompt.

משם זה התפתח ל-teammate שזמין כל הזמן. אפשר לתייג אותו כדי לענות על שאלה, לברר משהו על הארכיטקטורה, לעשות code review (בהומור: "it looks good to me", כמו מהנדס בכיר) או לקבל משימות מ-Linear. הדובר השני משתמש בשקופית "Introducing Claude Tag" כדי להתבדח שהם בנו את זה "six months before Anthropic released it", ומוסיף מיד שזה לא הנושא של ההרצאה.

Agent Chip...

Authors **>10%** of all merged PRs across tech

Runs **more than 1,800 tasks** every day

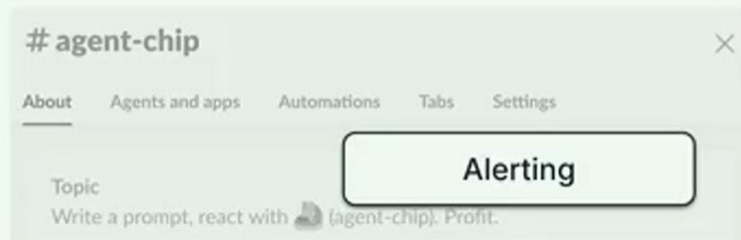
Is also actively adopted by **business users**



המדדים: Agent Chip כותב יותר מ-10% מה-PRs הממוזגים, מריץ יותר מ-1,800 משימות ביום ומאומץ גם על ידי business users

3. LLM = Lots of Little Multipliers

In January 2026 this went viral at Monzo...

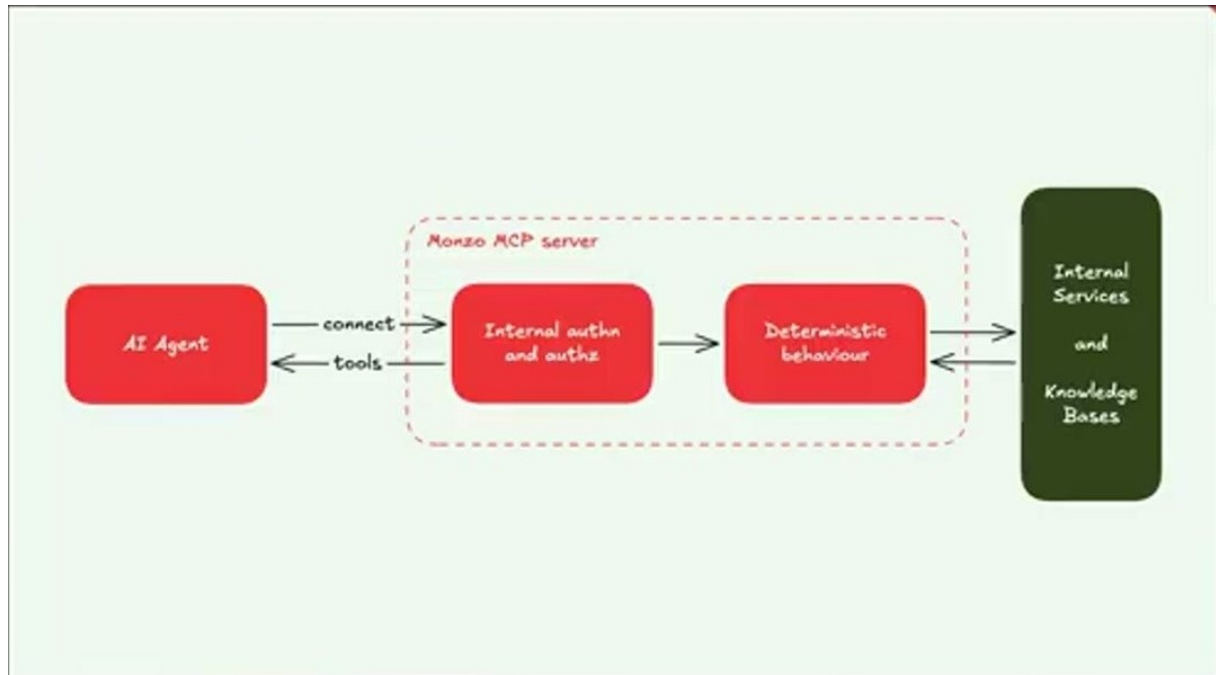


השקופית מ-Slack חוזרת, והפעם הכיתוב Alerting מודגש: כל רכיב בזרימה "react and profit" הוא יכולת פלטפורמה שכבר הייתה קיימת

זה הטיעון המרכזי. ה-"secret sauce" הוא LLMs, אבל "lots of little" LLMs, in this case, stands for multipliers. הזרימה הפשוטה (prompt, react, profit) נשענת על יכולות ש-Monzo כבר בנתה: alerting, monitoring, אינטגרציות עם צדדים שלישיים, סביבת staging שקל לבדוק בה, אבסטרקציות של queuing ו-database deployment פשוט.

המסקנה למי שמתכנן לבנות סוכן משלו: אם התשתית הזאת לא קיימת, הסוכן לא יהיה ההתחלה. הוא יהיה פרויקט platform שלם.

4. ה-MCP gateway וה-sandbox



דיאגרמת ה-MCP server: AI Agent מתחבר ומקבל tools, עובר דרך שכבת internal authn/authz ושכבת deterministic behavior, ומגיע ל-Internal Services ו-Knowledge Bases

עוד לפני שהיה סוכן מרוחק, Monzo השקיעה בתמיכה ב-coding agents מקומיים. הצעד הראשון היה MCP gateway לגישה למקורות ידע, לכלים פנימיים ולשירותים. ה-gateway נבנה כדי "to get the deterministic behavior, choosing what are the things we want to expose to the agents" והוא מנהל גם authentication ו-authorization לסוכנים השונים. ההנחה שמאחוריו: ככל שנותנים לסוכן יותר הקשר ונתונים בצורה בטוחה, כך התוצאה טובה יותר.

— **Felippe Felisola Caso (ייוחס משוער)** Again, we are a bank, we live in a regulated environment, so we want to make sure that the non-determinism from the agents is controlled

מעל ה-gateway נבנתה מערכת ה-agent orchestration שמפעילה את Agent Chip. הארכיטקטורה בנויה בשכבות:

namespace מבודד, מופרד משאר תשתית ה-production, שבו עולים pods עם sandboxes.

ה-agent image מגדיר בדיוק אילו כלים זמינים: סביבה, authentication, CLIs, ו-authorization שמגיעים מה-gateway.

proxy מחבר לעולם החיצוני, למשל ל-repositories של הקוד, אבל "we only get the egress that we choose to do, choose to give it".

בפנים רץ הסוכן עצמו, "that box inside the box inside the box is where the non-determinism happens". שם הוא חופשי לעשות מה שהוא רוצה. המטרה, כלשונם, היא לשלוט באי-הדטרמיניזם ובאותו זמן לתת לסוכן סביבה שבה הוא יכול "run loose and develop".

5. הסוכן כאבן בניין

ברגע שה-MCP וה-orchestration קיימים, Slack, GitHub, Linear ו-front-ends מתחברים כ-front-ends, ובפוטנציאל גם יותר מ-3,000 השירותים. הדובר מתאר את התוצאה העסקית: יותר משימות ומגוון רחב יותר שלהן, כי "we don't need people to be babysitting their agents all the time". מהנדסים ממשיכים לעבוד עם coding assistant מקומי, ובמקביל מאצילים משימות לסוכנים מרוחקים.

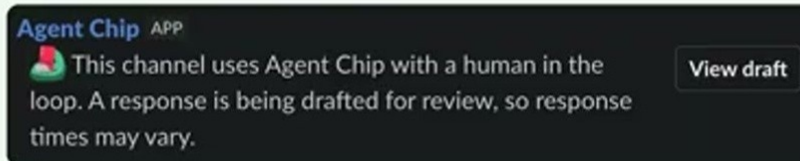
Working in incident mode and leveraging the MCP...



Agent Chip ב-incident mode: הודעה Incident Responder activated, ולצידה רשימת פעולות (קריאת הודעות, חקירה דרך logs, metrics, traces ו-config changes, סיכום חי, ממצאים עם מקורות, draft PR כשיש תיקון בביטחון גבוה)

Incident mode: כשנפתח Agent Chip incident, מצטרף לערוץ, חוקר logs ו-traces, מתחבר לשירות הרלוונטי ולקוד, ומספק לבני האדם מידע בעל ערך גבוה. ההחלטה העיצובית החשובה כאן היא ריסון: "We don't want the agent to be saying a lot of things during an incident and testing hypothesis life בכמות המעורבות שלו. לפי השקופית (הטקסט קטן וקשה לקריאה), הוא ממשיך לנטר עד 60 דקות או עד שמישהו כותב "incident resolved".

In #squad-ask channels on Slack...



ערוץ #squad-ask: הודעת Agent Chip שמבהירה שהערוץ עובד עם human in the loop, שהתשובה בטייטה ומחכה ל-review, ושיש כפתור View draft

ערוצי ask: אלה ערוצים שבהם squad אחד שואל squad אחר על שירות, על מוצר או על ה-Agent Chip roadmap. מנסח טייטה ל-human in the loop. לפעמים התשובה שלו מספיקה וסוגרת את הפנייה, ולפעמים הוא מבקש עוד הקשר ומתייג אדם.

Agent Chip On Call: בגרסה התגובתית הוא ממיין התראות על איכות נתונים ופותח חקירה אוטומטית. בגרסה היזומה הוא מנטר שינויים ב-backend ומזהה השפעה אפשרית על הנתונים "even before the data is processed". "in the next batch execution

אנשים שאינם מהנדסים: על סמך פידבק או רעיון הם קוראים ל-Agent Chip, מקבלים טיוטה שכבר נבדקה ב-"staging, and then, and only then, involve an engineer". ההדגשה של הדובר: "this doesn't change the rigor to which we develop software at Monzo".

6. למה להמשיך לבנות in-house

We believe Agent Chip is now a key part of our platform:

Its **composable** nature enables new use cases

The platform is **provider-agnostic** by definition

We're moving to cover a **bigger development window**

שלוש סיבות להמשיך: האופי ה-composable מאפשר שימושים חדשים, הפלטפורמה provider-agnostic מעצם הגדרתה, ו-Monzo מרחיבה את חלון הפיתוח שהסוכן מכסה

השוק zz מהר, ולכן נשאלת השאלה למה לא לקנות מוצר מוכן. התשובה של Monzo מורכבת משלושה חלקים: הראשון הוא composability: הסוכן הוא כבר אבן בניין שמחוברת לדרישות האבטחה והפלטפורמה. השני הוא אי-תלות בספק: "the little blue-slash-green agent box there can be very easily swapped for any model", ואפשר לבחור מודל באופן דינמי לפי המשימה. השלישי הוא הרחבת הכיסוי מקצה לקצה: טיוטות ל-backend-ול-app, בדיקות ב-"Maybe even predictors, staging, data tests."

הם גם לא נשמעים דוגמטיים: "even if we stop development today, this would already be valuable".
ובאותו זמן "Doesn't mean that we are necessarily married to agentship until the end".

7. מה לעשות עם זה

- **למפות קודם את ה-multipliers:** לפני שבונים סוכן, כדאי לבדוק אם קיימים staging, deployment פשוט, אינטגרציות ו-observability. אם לא, שם מתחילים.
- **MCP gateway לפני סוכן מרוחק:** שכבה אחת שמחליטה בצורה דטרמיניסטית מה חשוף ומנהלת auth משרתת גם סוכנים מקומיים וגם מרוחקים.
- **לבודד בשכבות:** namespace מבודד, image שמגדיר את הכלים, proxy עם egress מפורש, והסוכן רץ בקופסה הפנימית.
- **נקודת כניסה בלי חיכוך:** emoji או mention ב-Slack הם מה שהפך את הכלי לווייראלי. אל תבנו UI נפרד אם אין צורך.
- **לחשוב על הסוכן כרכיב פלטפורמה:** incident, on-call ו-ask הם וריאציות על אותו בסיס, ולא מוצרים נפרדים.
- **human in the loop כברירת מחדל:** טיוטות, draft PRs, ומהנדס שנכנס רק אחרי staging. כך שומרים על הרף ההנדסי גם כשאנשים שאינם מהנדסים תורמים.

שאלה פתוחה	למה זה חשוב
איך נמדד ה-10% מה-PRs, ומה איכותם?	בלי revert rate או זמן review קשה להעריך את הערך האמיתי
כמה זה עולה למשימה, וכמה אנשים מתחזקים את הפלטפורמה?	זה הבסיס להחלטה אם לבנות או לקנות

שאלה פתוחה	למה זה חשוב
באילו מודלים ו-harness משתמשים בפועל?	provider-agnostic הוא עיקרון, אבל לא נאמר מה רץ היום
איך מונעים זליגת מידע רגיש דרך ה-egress וה-proxy?	בבנק זו השאלה הרגולטורית המרכזית
מה עושה הגרסה היזומה של On Call כשהיא טועה?	false positives יכולים לשחוק את האמון בסוכן

8. מילון מונחים

מונח	פירוש
Coding agent / harness	סוכן AI שכותב ומריץ קוד, והתשתית שעוטפת אותו
MCP gateway	שרת Model Context Protocol מרכזי שחושף לסוכנים כלים ומקורות ידע נבחרים
Agent orchestration	מערכת שמפעילה סוכנים, מנהלת אותם וקוראת את התוצרים שלהם
Sandbox / isolated namespace	סביבת ריצה מבודדת מה-production
Egress	תעבורה יוצאת מהסביבה, כאן מוגבלת ליעדים שהוגדרו במפורש
Proxy	רכיב ביניים ששולט במה שהסוכן יכול להגיע אליו מבחוץ
Staging	סביבת בדיקה שקודמת ל-production
Incident mode	מצב שבו הסוכן מצטרף לערוץ incident וחוקר את התקלה
Human in the loop	אדם שמאשר או משלים את מה שהסוכן ניסח
Provider-agnostic	לא תלוי בספק מודלים מסוים
Data quality alert	התראה על כשל או חריגה בנתונים ב-data warehouse
Lots of Little Multipliers	המשחק המילים של ההרצאה: הרבה יכולות פלטפורמה קטנות שמכפילות את הערך