יסודות הדאטה כתשתית ל-AI

TLV-ANT201 — סיכום כנס, AWS Summit Tel Aviv 2026

Dan Khrakovsky (Sr GTM Specialist, Data & Analytics, AWS) · Ran Pergamin (Sr Solutions Architect Specialist, Data & AI, AWS) · Yaron Tomer (Director of R&D, Cloudinary)

10 בספטמבר 2026 | משך: כ-40 דקות | הופק מהקלטת הכנס

סיכום מאת קובי אלמוג

על המסמך הזה

המסמך מסכם את סשן הפתיחה של ה-Data & Analytics Track ב-AWS Summit Tel Aviv 2026, בהתבסס על ההקלטה המלאה ועל הסליידים שחולצו מהווידאו. הסשן מחולק לשלושה חלקים: חלק אסטרטגי (Dan Khrakovsky), חלק טכנולוגי על Open Data Architecture ועל העולם הסמנטי (Ran Pergamin), וסיפור לקוח מפורט של Cloudinary (Yaron Tomer). חותמות הזמן בגוף המסמך מפנות לנקודה המדויקת בהקלטה.

איך לקרוא את זה

- **פרק 1 — תקציר מנהלים:** חמש דקות קריאה. מה נטען בסשן ומה רלוונטי לארגון שמריץ דאטה בסקייל.
- **פרקים 2–4 — האסטרטגיה:** מה זה Data Strategy, למה Agentic AI חושף את חובות הדאטה, ושלושת הפילרים.
- **פרקים 5–7 — הטכנולוגיה:** Open Data Architecture, Apache Iceberg ו-S3 Tables, והשכבה הסמנטית והווקטורית.
- **פרק 8 — סיפור Cloudinary:** SnowChat — סוכן שאילתות בשפה טבעית מעל Snowflake, כולל מספרים ותוצאות.
- **פרק 9 — מה לוקחים מכאן:** מסקנות ונקודות להמשך.

הערת מקור המסמך הופק מתמלול אוטומטי של ההקלטה. התמלול טוב בעברית, אך מונחים לועזיים מופיעים בו בשיבוש פונטי (למשל "אפאצ'י אייסברג" = Apache Iceberg, "אג'נטיק" = Agentic) והם נורמלו כאן לכתיב האנגלי המקובל. הציטוטים מובאים כלשונם בהקלטה ואומתו מול חותמת הזמן המצוינת; מספרים שמופיעים על הסליידים צוינו ככאלה.

שקופית הפתיחה: שלושת הדוברים ותפקידיהם

1. תקציר מנהלים

הסשן פתח את מסלול הדאטה והאנליטיקה של הסאמיט, והטענה המרכזית שלו פשוטה: המחסום למעבר של Agentic AI לפרודקשן הוא לא המודל אלא הדאטה. משם הוא יורד לשתי שכבות מימוש — ארכיטקטורת דאטה פתוחה מעל Apache Iceberg, ושכבה סמנטית/וקטורית — ונחתם בסיפור לקוח שמראה איך זה נראה בפועל.

- **AI הוא צרכן דאטה מסוג חדש.** הפתיחה קבעה ש"אנחנו בעצם מבינים שאיי הוא צרכן שונה לחלוטין של דאטה מכל מה שעבדנו איתו עד היום" [0:00]; סוכן ניגש לדאטה איטרטיבית, אוטונומית, ובלי אדם שמגשר על הפערים.
- **Data Strategy זה לא רשימת פלטפורמות.** רוב הלקוחות, לדברי הדובר, מתארים תשתית כשנשאלים על אסטרטגיה. "Data Strategy אמיתי צריך לענות על שאלה אחת בסיסית מה הערך העסקי שהדאטה שלכם יכול לתת?" [1:35]
- **בעיית הדאטה אינה חדשה — Agentic AI רק הגדיל אותה.** סיילואים, מטא-דאטה לקוי וחוסר בבדיקות איכות היו תמיד; ההבדל הוא שהגורם האנושי שידע לעקוף אותם נעלם, וכעת הפער מתעצם "במהירות מכונה" [4:44].
- **היחידה התפעולית היא Data Product עם בעלים.** "יחידת דאטה, שיכולה להיות ממקור אחד או יותר, אבל יותר חשוב מזה, יש לה אונר" [3:12] — בעלים שאחראי להקשר העסקי, לבדיקות האיכות ולהנגשה לארגון.
- **השכבה הטכנית: פורמט טבלה פתוח + שכבה סמנטית.** Apache Iceberg כשפה משותפת לכל המנועים, S3 Tables כשירות מנוהל שמטפל ב-compaction ובתחזוקה, ולצידם OpenSearch ו-Amazon S3 Vectors לשכבה הווקטורית [16:51], [23:00].
- **Cloudinary הראו שזה עובד, ומדדו את זה.** SnowChat — MCP server מעל Snowflake — הוריד את צוואר הבקבוק של האנליסטים, ושני סיפורי הצלחה הוצגו: שיפור של 52% ב-activation rate בשלושה חודשים [35:22], וגילוי עצמאי של 4,800 משתמשי free tier שחורגים מהמכסה [36:54].

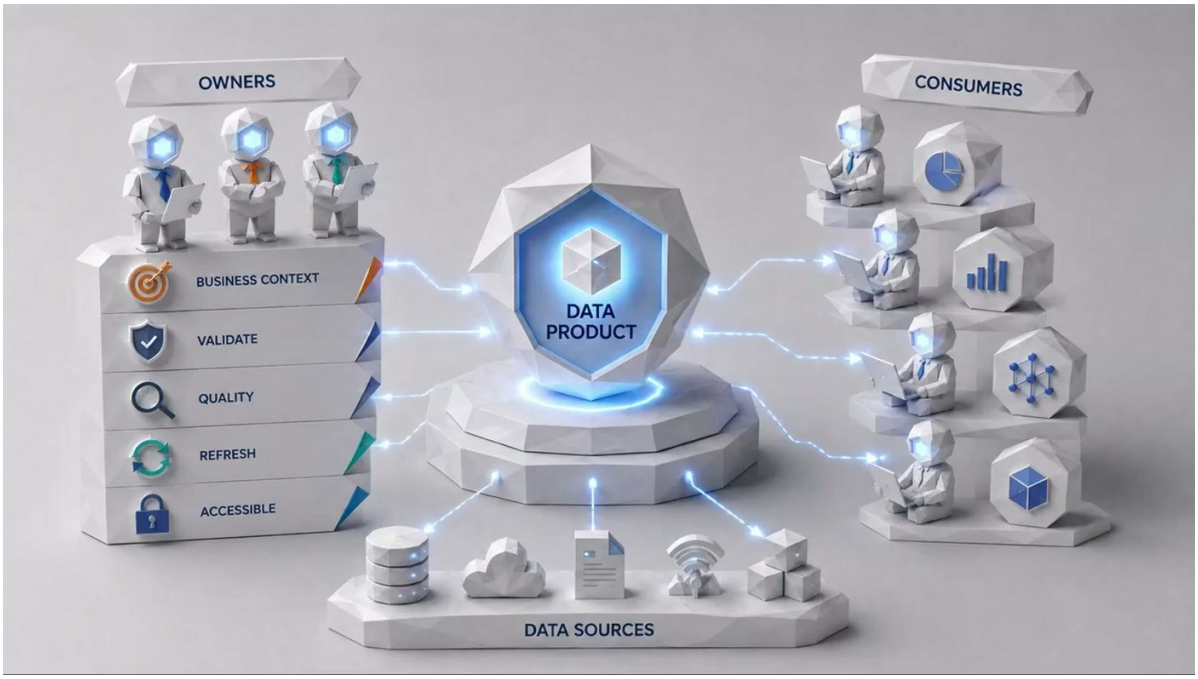
2. מהי אסטרטגיית דאטה, ומדוע רוב הארגונים לא מחזיקים באחת

הסשן נפתח בשאלה לקהל: כמה הריצו POC אג'נטי בשנתיים האחרונות, וכמה הגיעו לפרודקשן. לפי תיאור הדובר, מספר הידיים בשאלה השנייה היה קטן משמעותית — וזה שימש נקודת המוצא לכל מה שבא אחריו.

הדובר תיאר דפוס חוזר בשיחות עם לקוחות: כששואלים אותם מה אסטרטגיית הדאטה שלהם, מקבלים תיאור של פלטפורמה — שמות של תשתיות ופתרונות — ולא תשובה עסקית. לטענתו זה לא אסטרטגיה אלא "במקרה הטוב איזשהו דאטה פרוג'קט" [1:35].

> **Dan Khrakovsky, [1:35]** אמיתי צריך לענות על שאלה אחת בסיסית מה הערך העסקי שהדאטה שלכם יכול לתת? — Data Strategy

מכאן נגזרת ההגדרה המעשית שהוצגה: כשסוכן צריך דאטה, לא מספיק להצביע על dataset או על warehouse. הסוכן צריך דאטה שזמין לו, שהוא יכול לסמוך על אמינותו, ושהוא יכול לבצע עליו פעולות — בצורה אוטונומית. היחידה שעונה על כל אלה נקראה Data Product.



Data Product במרכז: מצד אחד בעלים שאחראים על הקשר עסקי, ולידציה, איכות, רענון ונגישות; מהצד השני צרכנים — אפליקציות, BI וסוכנים

לצד ההגדרה הוצגו שני מספרים מדוח אנליסטים (כפי שנאמרו מהבמה, [3:12]): עד 2027, 50% מהחברות שמריצות היום GenAI ישתמשו בסוכנים כחלק מהמערכות שלהן; ועד 2028, 33% מהחברות שמפתחות תוכנה ישתמשו בסוכנים בתהליך הפיתוח — לעומת אחוז אחד ב-2024.

3. למה Agentic AI חושף את חוב הדאטה

הטענה המרכזית של החלק הזה היא שהמעבר לפרודקשן נכשל לא בגלל הטכנולוגיה. סביבה מבוקרת עם דאטה סינתטי היא סיפור אחד; סקייל של פרודקשן, עלויות וסיכון הם סיפור אחר.

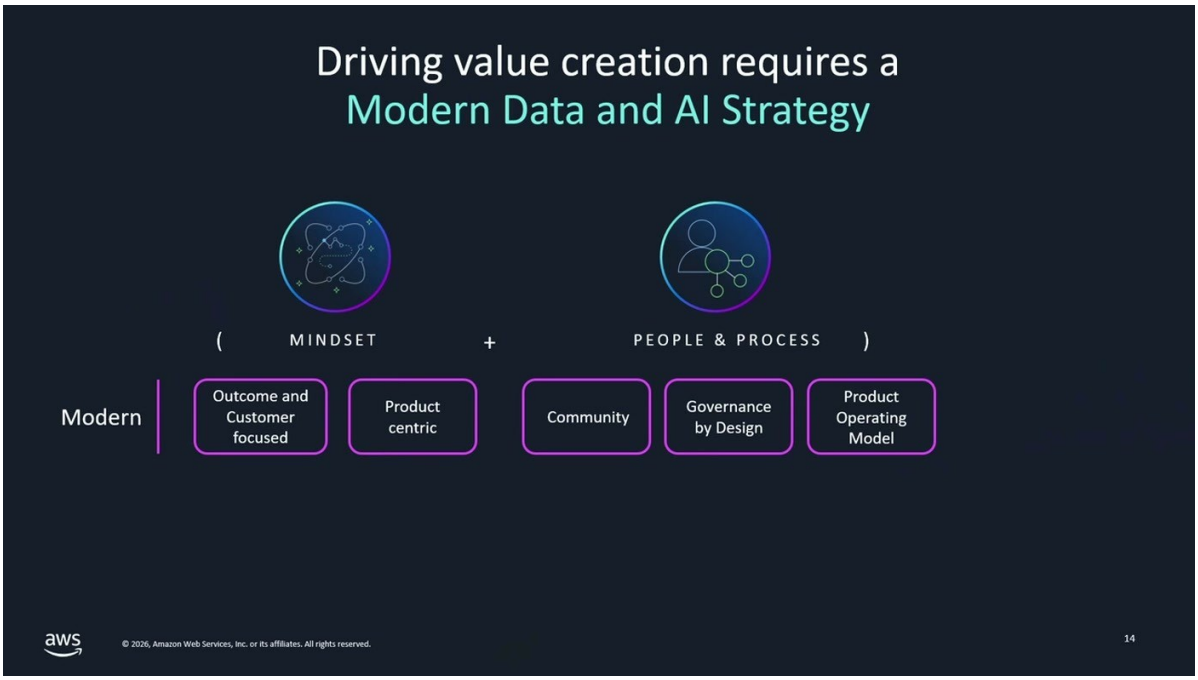
> **Dan Khrakovsky, [3:12]** הבעיה היא לא בהכרח בעיה טכנולוגית. היא קודם כל בעיה של דאטה. —

מה שחדש הוא לא הבעיה אלא החשיפה שלה. "שנים אנחנו עבדנו עם דאטה שהוא בסיילו, שנים עבדנו עם דאטה שהוא חסר בבדיקות איכות או מתה דאטה לקוי" [4:44] — וכל אלה נספגו על ידי הגורם האנושי שידע לגשר עליהם. סוכן אוטונומי לא מגשר; הוא מכפיל את הפער.

- **עלות.** כל דאטה שיושב בסיילו וצריך להיות מועתק עבור מערכות ה-AI מעלה עלויות. סוכן לא מבצע query אחד — התהליך איטרטיבי, ולכן כל גישה לדאטה לא ממודל או לא אופטימלי מתומחרת שוב ושוב.
- **לייטנסי.** סוכן שלוקח לו זמן בלתי סביר להגיע ל-dataset שלו הוא לא מצב תקין, גם אם התשובה בסוף נכונה [4:44].
- **אמון והזיות.** הוצגה כבעיה שאיננה רק בעיית מודל: איפה שחסר הקשר, ולא ניתן להגדיר אותו, האמינות של המערכת יורדת [6:16].
- **איכות.** "data שלא עובר בדיקות איכות וdata שהוא לא עקבי בסופו של דבר מביא מערכות AI לקבלת החלטות שגויה" [6:16].

המסקנה שנוסחה מהבמה אסטרטגיית הדאטה היא ה-foundation לאסטרטגיית ה-AI — ולא פרויקט מקביל לה. זה המונח שתחתיו נבנה כל המסלול: Data Foundation for AI.

4. שלושת הפילרים: מיינדסט, אנשים ותהליכים, טכנולוגיה



שני הפילרים הראשונים כפי שהוצגו על השקף: מיינדסט (מיקוד בתוצאה ובלקוח, חשיבה מוצרית) ואנשים ותהליכים (קהילה, ממשל מתוכנן, מודל תפעולי מוצרי)

- **מיינדסט.** ארגון מסורתי בוחר פלטפורמה ואז מחפש את הבעיה שהיא פותרת. ארגון מודרני מתחיל מהערך העסקי, ומגדיר בעלים, roadmap ו-KPIs. הניסוח מהבמה: המחשבה עוברת "ממה הכלי יכול לעשות, למה הלקוח שלי צריך" [6:16].
- **אנשים ותהליכים — כאן נתקעים.** זה הפילר שבו, לפי הדובר, רוב הארגונים נעצרים: צוות AI מרכזי אחד, מנותק מהצד העסקי, בלי דמוקרטיזציה של מידע. החלופה היא צוותים cross-functional, אג'ליים ומבוזרים [7:46].
- **טכנולוגיה — אחרונה בסדר.** מונוליט סגור עם נעילת ספק מול Open Data Architecture מודולרית וגמישה — הנושא שאליו הוקדש החלק השני של הסשן [7:46].

> **Dan Khrakovsky, [7:46] —** דמוקרטיזציה של מידע אומרת, לא שכל אחד ניגש לכל מידע שהוא יש, אלא כל אחד יכול לגשת למידע שהוא צריך.

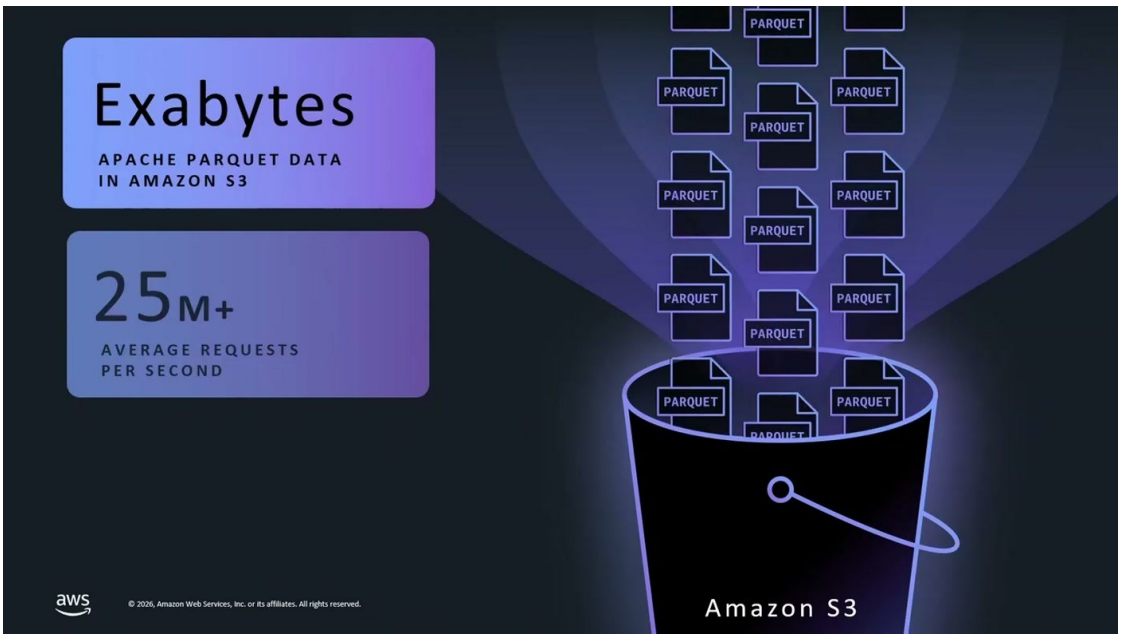
הארגון שמאמץ את שלושת הפילרים מגיע למה שהוצג כ-Modern Data Community: יצרני דאטה שאחראים על ההקשר ועל הפרסום, צרכני דאטה (אפליקציות, צוותי פיתוח, מערכות AI), וטכנולוגיה באמצע שמאפשרת את החילופין — כאשר התפקידים מתחלפים.

האנלוגיה שהוצגה הייתה סופרמרקט: כל המוצרים מקוטלגים, הלקוח נכנס, מוצא את מה שהוא צריך, ומגיע לקופה. הפרודיוסרים — מפעילי הסופר — רואים מה נצרך ומה לא, והופכים בעצמם לצרכני מידע. לעומת זאת, הוצג תרחיש היפותטי שבו יש "סלקטור בכניסה", כולם עומדים בתור, מבקשים ממנו מוצר, והוא הולך לחפש — לפעמים מוצא ולפעמים לא [10:48]. זו, לפי הדובר, התנהלות הדאטה בפועל בלא מעט ארגונים.

5. Open Data Architecture: שפה משותפת לכל המנועים

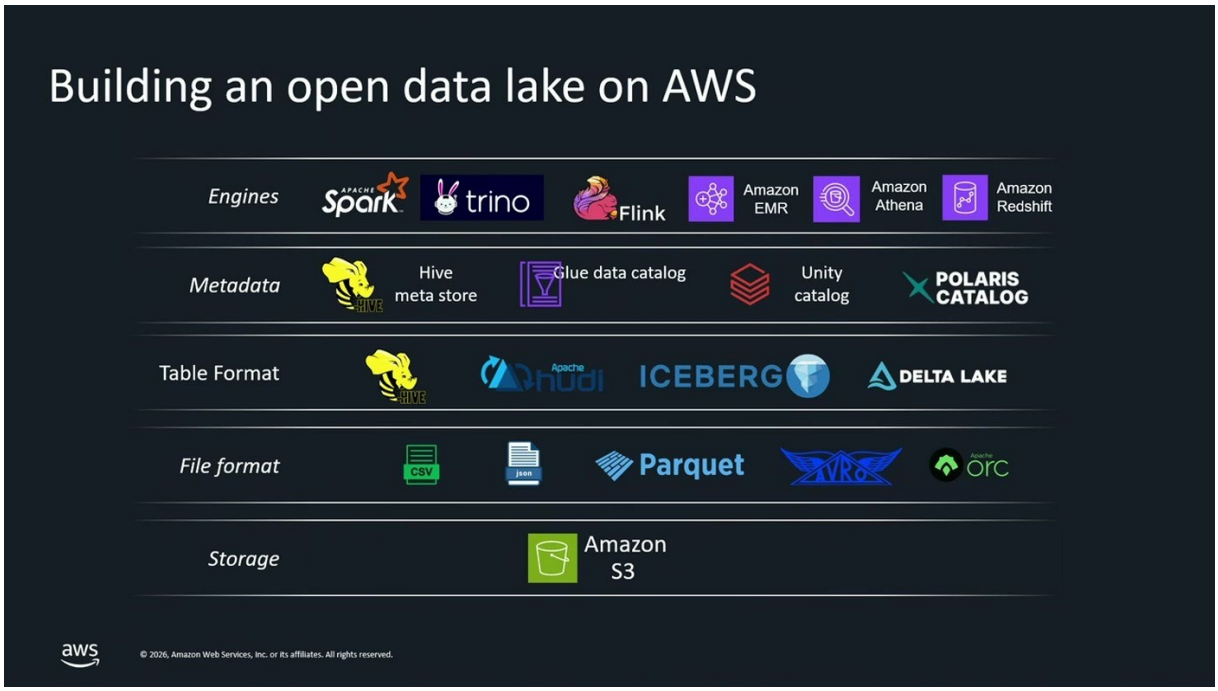
החלק הטכנולוגי נפתח בהגדרה מינימלית של מה זו ארכיטקטורה פתוחה.

> **Ran Pergamin, [12:19] —** כשאני רוצה לעבוד עם פרודוסרים וקונסומרים ואני רוצה שכולם יגישו לאותו דאטה יקבלו את אותה איכות ימצאו את מה שהם רוצים הם כולם צריכים לדבר שפה משותפת וזה המשמעות הבסיסית של Open Architecture



המספרים שהוצגו על נפח ה-Parquet ב-Amazon S3: אקסה-בייטים של דאטה, ולפי השקף 25M+ בקשות בממוצע בשנייה

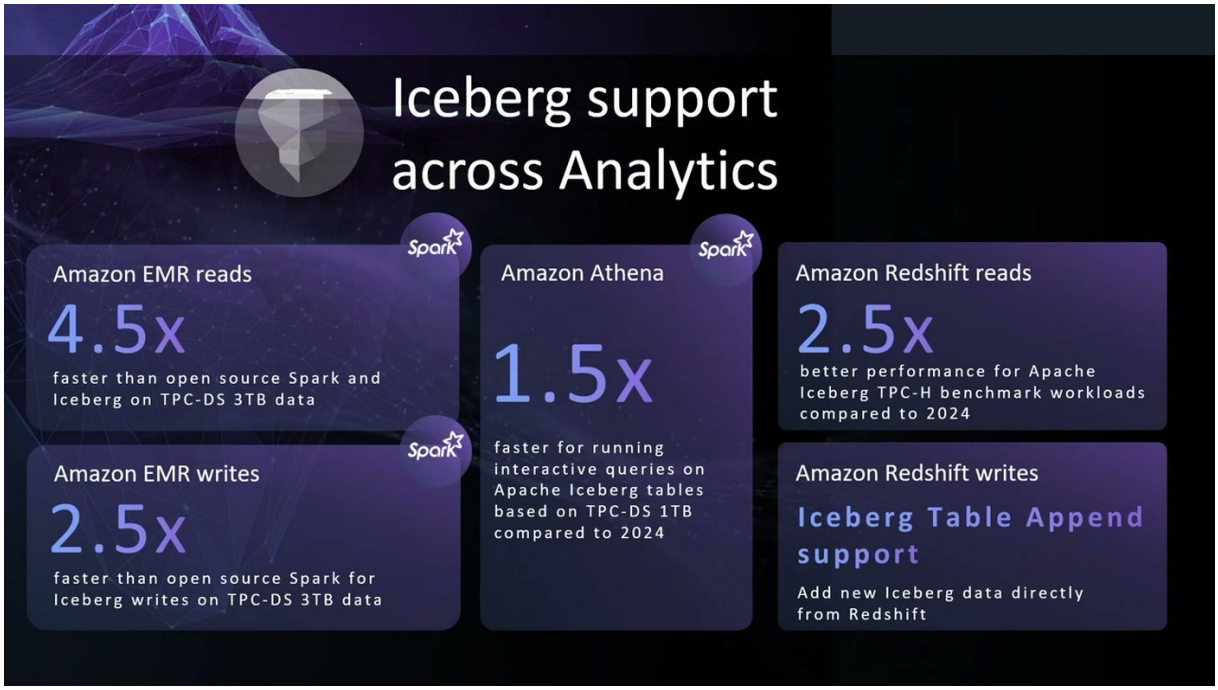
הרקע ההיסטורי שהוצג: Data Lake נולד כדי לבצע decoupling בין compute ל-storage ולשמור מידע טבלאי בסקייל גדול — תחילה on-prem, אחר כך בענן מעל S3. אבל קבצים לבדם (Parquet, Avro, CSV, JSON) לא מספיקים כדי לייצג טבלאות בסדרי גודל של פטה-בייט; צריך שכבה שמארגנת אותם למבנה לוגי, ומעליה קטלוג שמשמש נקודת כניסה — מה נמצא איפה, ומי ניגש למה.



הסטאק הפתוח שכבה-שכבה: מנועים, מטא-דאטה וקטלוגים, פורמט טבלה (Iceberg, Delta Lake, Hudi), פורמט קובץ (Parquet, ORC) ואחסון על S3

האבולוציה מ-Data Lake ל-Lakehouse נועדה להביא יכולות של בסיס נתונים אל הדאטה שיושב ב-S3: SQL, Time Travel, Schema Evolution ו-Partition Evolution — דברים שקשה מאוד לעשות מעל Hive והפורמטים המוקדמים. AWS בחרה ב-Apache Iceberg כפורמט הטבלה שעליו נשען רוב מה שנעשה היום בעולמות ה-Data Lake וה-Lakehouse.

הרעיון האחד שכדאי לזכור "הפרידו את שכבת המטה-דאטה מהדאטה" [15:21]. זו ההפרדה שמאפשרת לתכנן מראש במה לגעת ואיך, ומשם נגזרים Time Travel, Schema Evolution, Partition Evolution ו-Predicate Pushdown. בניסוח הדובר: "זה הדבר המהותי ביותר שהאופן טייבל פורמאטס והאפאצי אייסברג עושים" [15:21].



מספרי הביצועים שהוצגו לאינטגרציה של Iceberg בשירותי AWS (כפי שמופיעים על השקף)

ההשקעה באינטגרציה הוצגה כרוחבית: EMR, Athena ו-Redshift מכוונים לעבוד מול Iceberg בביצועים טובים יותר, לצד היכולת להתחבר גם למנועים חיצוניים כמו Snowflake ו-Databricks [16:51].

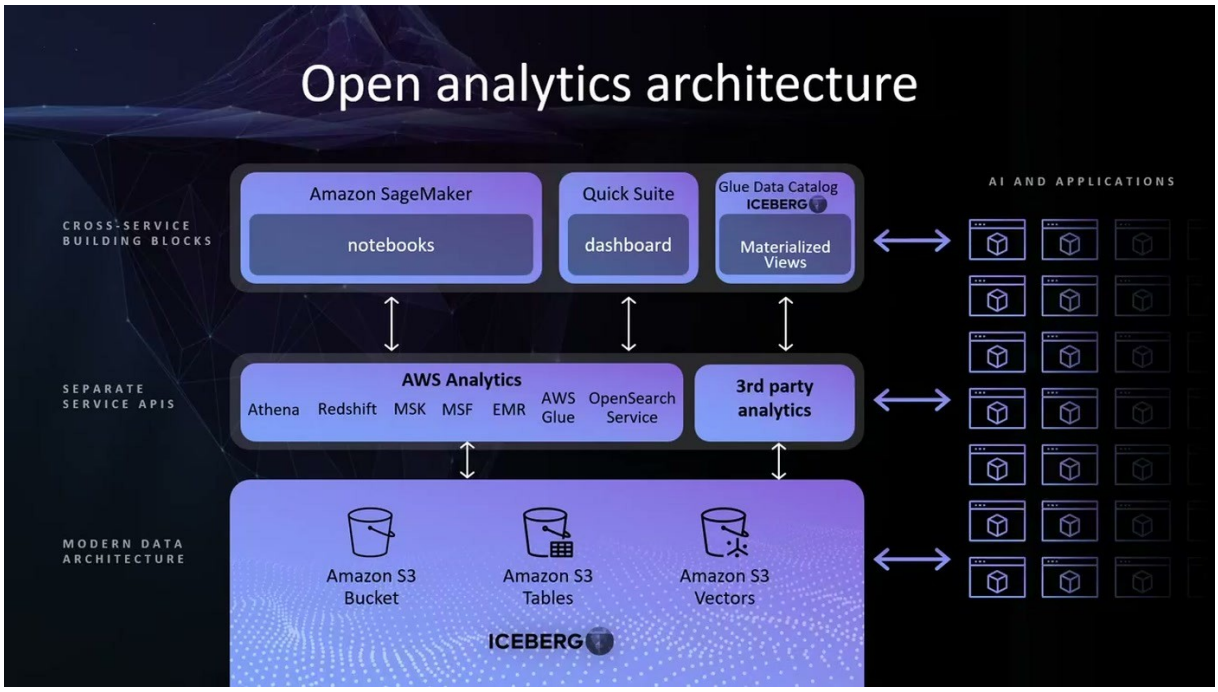
6. S3 Tables: Iceberg מנוהל, ובעיקר תחזוקה מנוהלת

> **Ran Pergamin, [16:51]** — אפאצ'י אייסברג זה לא פלטפורמה

הנקודה הזו היא שורש הבעיה שהוצגה: Iceberg הוא מפרט וסט ספריות שמגדיר איך הדאטה מאוחסן, כך שכל המנועים ידברו את אותה שפה. אבל אין בו מנוע שמנהל את הדאטה בפועל — וזה משאיר את העבודה הקשה על הצרכן.

S3 Tables כ-bucket מסוג חדש: Iceberg מנוהל שכל כלי תואם-Iceberg יכול לתשאל

Amazon S3 Tables הוצג כשירות ב-data plane של S3 שמציע Apache Iceberg מנוהל. הודגש שזה לא "קבצים ב-S3 ומשהו אחר שמסתכל עליהם" (הדפוס המוכר מ-Glue) אלא bucket מסוג חדש. הערך המרכזי הוא בדיוק הדברים הקשים לתפעול [18:23]: maintenance, ניקוי קבצים מיותרים, compaction, וסידור מחדש של דאטה שנכנס ב-ingestion ולא נשמר בצורה אופטימלית ל-queries. המסר: שהצוות והמנועים יתרכזו ב-query וב-ingestion.



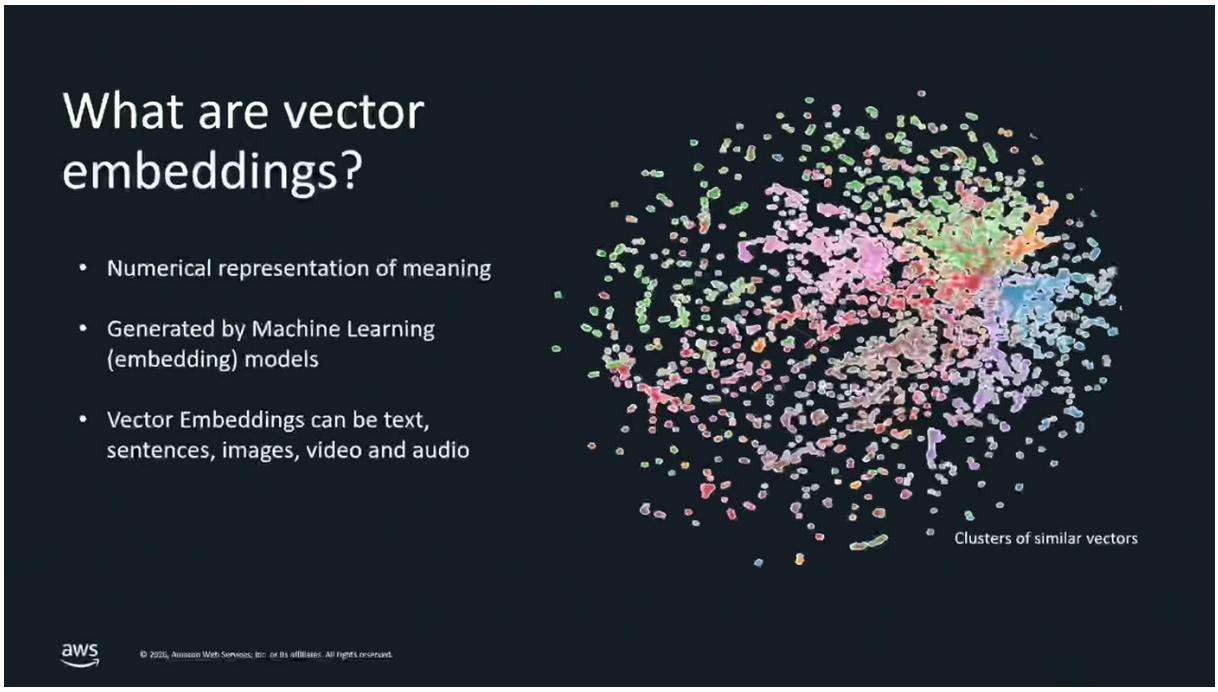
התמונה המלאה: S3 Bucket, S3 Tables ו-S3 Vectors בבסיס, מנועי AWS מעליהם, ומעליהם אפליקציות וסוכנים

נושא נוסף שקיבל דגש הוא materialized views — מענה לכך שלכל צרכן יש שאלה אחרת. הן מאפשרות להצליב טבלאות ולקבל מיד את החיתוך הנדרש, בלי שכל צרכן ירוץ מחדש על כל הדאטה [18:23].

7. השכבה הסמנטית: וקטורים, OpenSearch ו-S3 Vectors

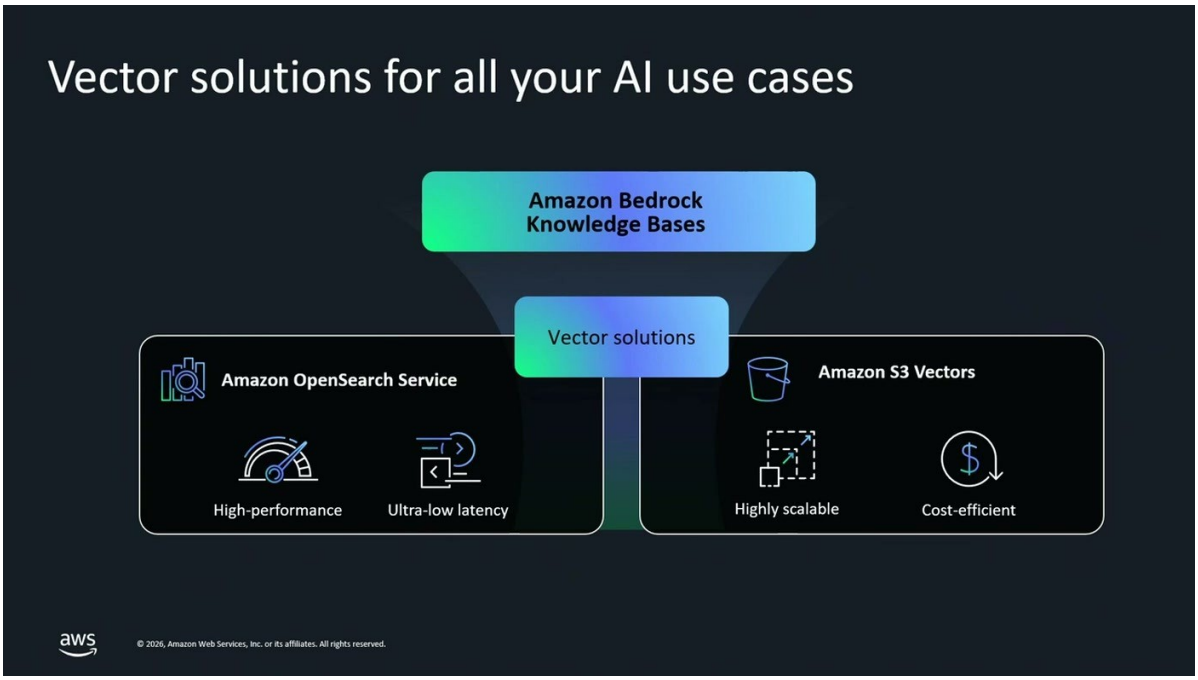
Ran Pergamin, [19:57] — ווקטורים הם השפה שאייג'נטים מדברים והיא שפה מתמטית

הרקע שהוצג: היסטורית חיפשנו מידע לקסיקוגרפית — טקסט, מילות מפתח, ומאוחר יותר טבלאות, עמודות ושורות, שגם הן עדיין לקסיקוגרפיות. הווקטורים נכנסו עם עולם הצ'טבוטים, והיום הסוכנים צריכים הרבה יותר מזה — הם צריכים הקשר, ובצורה שהיא סוג של דטרמיניסטית.



ההגדרה כפי שהוצגה: ייצוג מספרי של משמעות, שנוצר על ידי embedding model, ויכול להיות טקסט, תמונה, וידאו או אודיו

ההסבר ניתן דרך דוגמה: קופסה בשם embedding model — מצד אחד נכנס פריט מידע (תמונה, וידאו, פסקה), ומהצד השני יוצא מערך מספרים שמייצג אותו במרחב רב-מימדי. אחרי שמעבירים את כל השירים בעולם דרך הקופסה, ניתן לחפש לפי קרבה במרחב (vector proximity). הדוגמה שהמחישה את ההבדל: חיפוש "שירים לברביקיו אמריקאי או על-האש ישראלי אחר הצהריים משנות ה-80" — משימה קשה מאוד בחיפוש טקסטואלי או טבלאי, וטבעית במרחב וקטורי [21:27].



שתי הגישות זו לצד זו: OpenSearch לביצועים ולאטנסי נמוך במיוחד, S3 Vectors לסקייל גבוה ועלות נמוכה

שני השירותים לא הוצגו כתחליפים אלא כשתי גישות. כשהווקטורים היו מעטים, שמרו אותם בזיכרון ועל מדיה מהירה. בשנתיים האחרונות השתנו שני דברים: הסקייל (אינדוקס של כל האינטרנט, או של סרטוני YouTube בפירוק לדקות — טריליוני וקטורים), והציפייה ללייטנסי — לא כל תשובה חייבת להגיע במילישניות בודדות.

> **Ran Pergamin, [23:00]** נותן לי את הגישה המהירה מאוד ואמזון ה-3 וקטורס בא לתת הגישה — OpenSearch היותר cost efficient

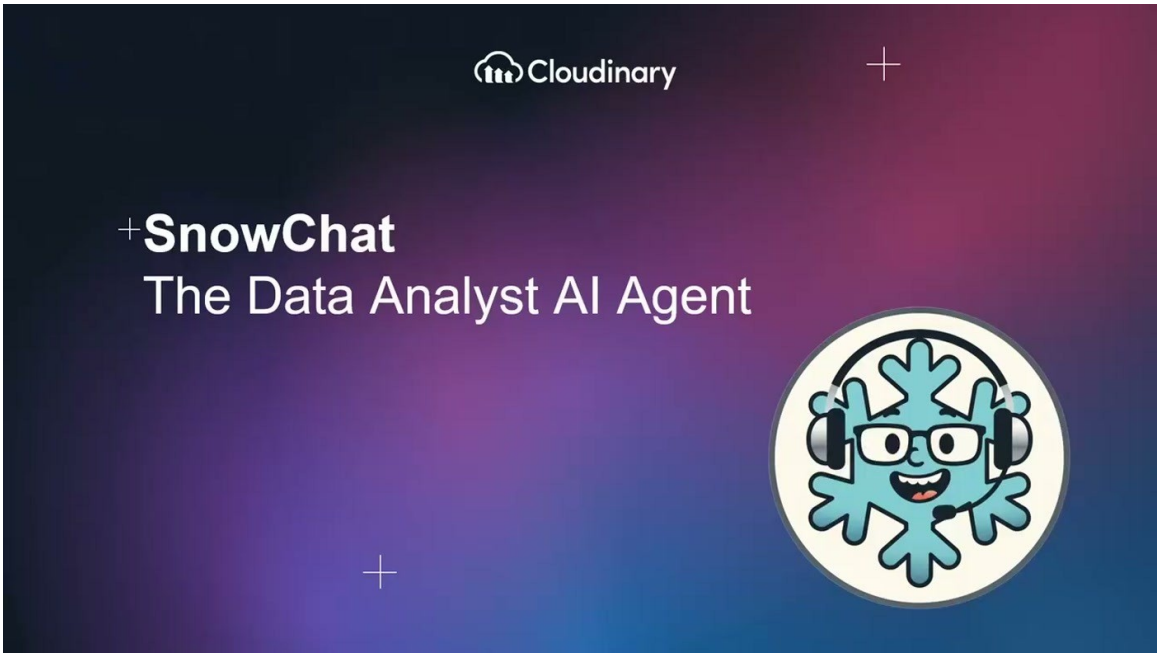
- **OpenSearch.** התחיל כמנוע לקסיקוגרפי והתפתח לפלטפורמת חיפוש מלאה שגם עולם הווקטורים נכנס אליה. מאחסן בזיכרון ובדיסקים מהירים ומצריך compute שרץ — וגם הרבה "כפתורים לכוון", כלומר בשלות טכנולוגית מהארגון [24:32], [26:04].
- **Amazon S3 Vectors.** הוכרז GA ב-re:Invent הקודם. bucket מסוג אחר, שנועד לאחסון וקטורים בסקייל של מאות מיליארדים ואף טריליונים, בעלות של S3. הלייטנסי הוגדר "good enough": "זה sub-second, כשזה 100 millisecond בערך לקואריז הכי מהירים שיש" [24:32].
- **האינטגרציה בין השניים — בשני הכיוונים.** מ-OpenSearch אפשר להוריד ל-S3 Vectors וקטורים שלא בשימוש תדיר כדי להוזיל עלות; ובכיוון ההפוך, מי שיושב על S3 Vectors יכול להעלות אינדקס ל-OpenSearch Serverless (הפעולה נקראה מהבמה elevation) ולהרים compute לפי הצורך — לאירוע, למבצע או למחקר — ולקבל את הלייטנסי הנדרש רק לפרק הזמן שצריך [26:04].

הסיכום של החלק הזה: העולם הסמנטי הוא "שווה זכויות וחובות" לצד הטבלאות, ושניהם יחד מאפשרים גם לאנשים וגם לסוכנים לגעת בדאטה במקום אחד ולקבל את התמונה הכוללת — וזו בדיוק הדרך לייצר data products [26:04].

8. Cloudinary: SnowChat — מכונת הקפה של הדאטה

> **Yaron Tomer, Cloudinary, [27:35] —** אני מרגיש שלהמתין לדאטה מדאטה אנליסט זה כמו להמתין לקפה מבריסטה

האנלוגיה שפתחה את סיפור הלקוח: בית קפה לא תמיד פתוח, עומדים בתור כדי להזמין, ואז עומדים בתור נוסף כדי לקבל — ובסוף מקבלים מוצר איכותי, אבל יש צוואר בקבוק. הפתרון המקביל הוא מכונת קפה: תמיד זמינה, כמה שרוצים ומתי שרוצים. המוצר שנבנה ב-Cloudinary נועד להיות מכונת הקפה של הדאטה — סוכן שכל אחד בארגון יכול לתשאל בשפה טבעית.



המוצר שנבנה בתוך Cloudinary: SnowChat, סוכן AI בתפקיד אנליסט הדאטה

הסקייל שמאחורי הסיפור

- **התשתית.** Cloudinary מנהלת, מטרנספרמת, מבצעת אופטימיזציה ומגישה מדיה עבור לקוחות כמו האתרים הגדולים שכולם מכירים — כולל התאמת פורמט לדפדפן (למשל AVIF מול JPEG) וחיתוך חכם.
- **נפח יומי.** "כמה תמונות? 1300 טראבייט של תמונות וסרטים כל יום" [29:12] — ומזה נוצרים לוגים של CDN בהיקף שדורש עיבוד יעודי.
- **הפייפליין.** Spark שרץ על EMR מבצע את העיכול, והתוצאה נכנסת ל-S3 בפורמט Iceberg. "זה מייצר 16 מיליארד רקורדס חדשים כל יום" [30:48].
- **הצריכה.** Snowflake קורא את הטבלאות ב-S3 דרך Glue [30:48] — בדיוק דפוס ה-Open Architecture שתואר בחלק הקודם.

הארכיטקטורה של SnowChat

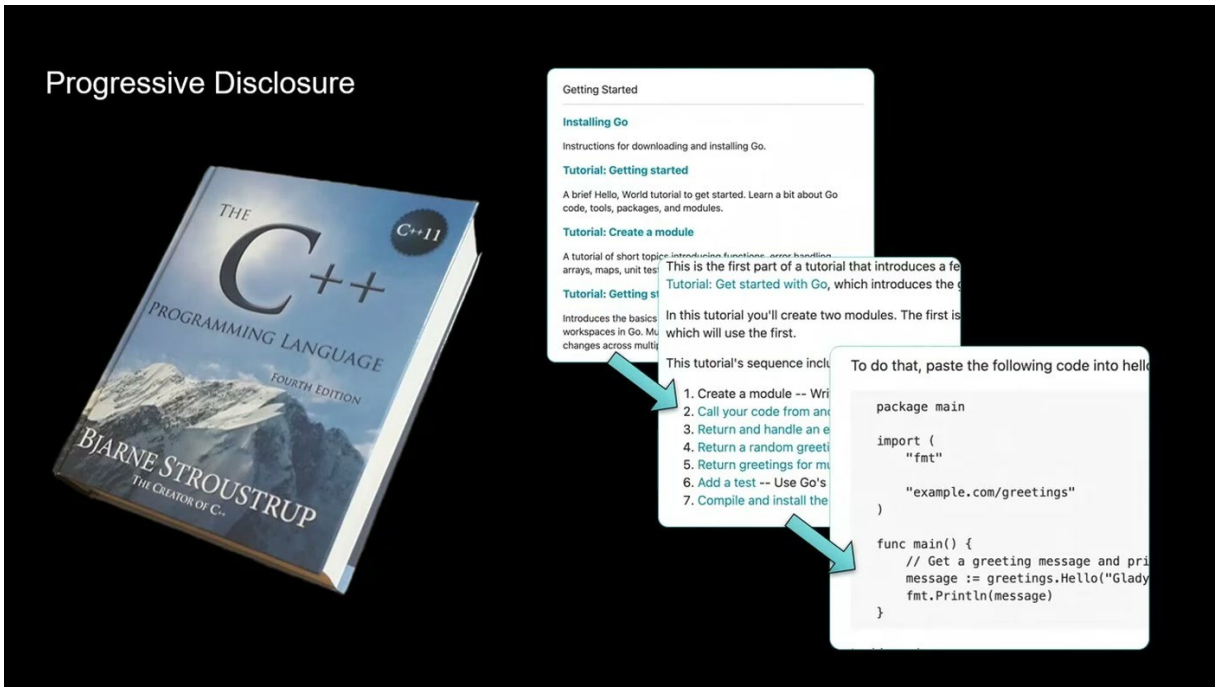
> **Yaron Tomer, [30:48] —** בסופו של דבר Snowchat זה MCP סרבר שרץ בענן. קלוד מתחבר אליו באמצעות MCP Remote, והוא נותן גישה לסנופלק דאטאבייס.

הדובר הדגיש מיד שזה לא מספיק: "בצורה נאיבית פה זה היה צריך להסתיים אבל זה ממש לא מספק" [30:48]. התוצאות לא היו איכותיות בלי שכבת ידע נוספת. השכבה הסמנטית שנבנתה מעל כללה ארבעה רכיבים [32:18]:

- **General instructions.** הנחיות כלליות לסוכן על אופן העבודה.
- **תיאור סכמות ועמודות.** מה התוכן בכל עמודה, ומתי להשתמש בה.
- **דוגמאות SQL.** כמו שדוגמאות עוזרות לבני אדם להבין מה מבקשים מהם, כך גם ל-LLM.
- **Lookups.** נוספו כדי לשפר את מהירות הריצה.

הבעיה שצצה בסקייל, והפתרון: Progressive Disclosure

הפתרון עבד טוב על מספר קטן של טבלאות, אבל כשעלו לכמות גדולה כל טבלה נוספת הקשתה על האיכות. המטרה שהוגדרה: שכל טבלה נוספת תוסיף רק עלות שולית קטנה. הניסיון הראשון היה vector embedding — ולפי הדובר, "וקטור אמבדינג במקרה שלנו לא היה מספיק טוב" [32:18]. משם עברו לשיטה אחרת.



האנלוגיה שהוצגה ל-Progressive Disclosure: במקום לקרוא ספר שלם — לפתוח עמוד, ללחוץ על הקישור הרלוונטי, ולהמשיך עד שיש מספיק מידע

> **Yaron Tomer, [32:18] —** במקום לקרוא את הכל, אנחנו נתחיל לקרוא פיסת מידע קטנה, ממנה נבין מה הפיסת מידע הנוספת שהקלוד צריך

שני מנגנוני האיכות

- **דירוג עצמי (confidence level).** "אבל אז אמרנו בעצם לאייג'נט בוא תנסה לדרג את עצמך, בוא תתן לעצמך איזשהו confidence level" [33:48]. זה מה שעשה את השינוי: כשה-confidence נמוך (medium ומטה), הסוכן לא מציג תוצאה אלא הודעה במקומה.
- **רגרסיה אוטומטית מול Snowflake.** "היינו צריכים לעשות גם בדיקות regression, בדיקות איכות" [33:48] — למעלה מ-100 test cases, שכל אחד מהם מורכב מ-prompt ומ-validation query. בזמן ריצה הטסט מזין את ה-prompt, SnowChat מייצר query משלו, שני ה-queries רצים ב-Snowflake, והתוצאות מושוות — ואמורות להיות זהות [35:22].

מה זה נתן

סיפור ההצלחה הראשון כפי שהוצג: עקומת ה-activation rate וגידול של 52% בשלושה חודשים

- **צוואר הבקבוק נפתר.** לפי הדובר, יש בארגון מי שכבר אומרים שאי אפשר לחזור אחורה; האנליסטים עצמם הרוויחו — הם מתמקדים בנושאים אסטרטגיים במקום בבקשות אד-הוק [35:22].
- **נוצרה שיטת data mining חדשה.** תוצאה שלא ציפו לה מראש: עבודה איטרטיבית של אנשי מוצר מול הדאטה [35:22].
- **Success story 1 — activation rate.** מנהל מוצר תשאל את SnowChat בעצמו, בנה ניסוי, וחזר על המחזור הזה איטרטיבית. "הוא הגיע למצב ששיפר ב-52% את האקטיביישן רייט שלנו" [35:22] — בשלושה חודשים.
- **Success story 2 — הסוכן מצא את זה לבד.** כאן לא נשאלה שאלה ספציפית אלא כללית: איך אפשר לשפר את ההכנסות. איטרטיבית, SnowChat חיפש מקורות הכנסה ועצר על ממצא: "מסתבר ש-4,800 מהם, למעשה, חורגים מעבר לפרי טיר" [36:54] — משתמשים שמנצלים את ה-grace period וצורכים יותר credits מהמתוכנן. סגירת הפרצה והפיכתם ללקוחות משלמים היא הכנסה נוספת.

> **Yaron Tomer, [36:54] —** אנחנו היינו מהחלוצים של הדבר הזה, התחלנו עם זה לפני שנה וחצי, זה לא היה קל

9. מה לוקחים מכאן

שקופית הסיכום שהוצגה: דמוקרטיזציה של דאטה — אינטראופרביליות, פירוק סיילואים, מזעור תזוזת דאטה

- **התחילו מהשאלה העסקית, לא מהפלטפורמה.** מבחן פשוט לארגון: אם התשובה לשאלה "מה אסטרטגיית הדאטה שלנו" היא רשימת כלים, אין אסטרטגיה.
- **הגדירו בעלות על Data Products.** לכל יחידת דאטה צריך בעלים שאחראי על ההקשר העסקי, על בדיקות האיכות ועל ההנגשה — אחרת הסוכן יירש את כל החובות.
- **פורמט טבלה פתוח הוא תנאי, לא אופנה.** Apache Iceberg כשפה משותפת מאפשר לכל המנועים — פנימיים וחיצוניים — לראות אותה תמונה. S3 Tables מוריד את עלות התחזוקה (compaction, ניקוי קבצים) מהצוות.
- **השכבה הסמנטית שווה במעמדה לשכבה הטבלאית.** בחרו בין OpenSearch ל-S3 Vectors לפי דרישת הלייטנסי והסקייל, ודעו שיש מעבר בין השניים בשני הכיוונים.
- **מ-Cloudinary — שלושה דפוסים ניתנים להעתקה.** Progressive Disclosure במקום לדחוף סכמה שלמה להקשר; confidence level שמונע הצגת תשובה לא בטוחה; וסוויטת רגרסיה של 100+ מקרים שמשווה תוצאות query מול validation query.
- **מדדו ערך עסקי, לא שימוש.** שני סיפורי ההצלחה שהוצגו נמדדו בשינוי עסקי (52% ב-activation rate, זיהוי 4,800 חשבונות חורגים) ולא במספר שאילתות.

נספח — מונחים שהוצגו בסשן

מונח	משמעות כפי שהוצגה
Data Product	יחידת דאטה ממקור אחד או יותר, עם בעלים אחראי להקשר, לאיכות ולהנגשה
Data Foundation for AI	התפיסה שאסטרטגיית הדאטה היא הבסיס לאסטרטגיית ה-AI ולא מקבילה לה
Modern Data Community	מודל של יצרני דאטה, צרכני דאטה וטכנולוגיה באמצע, שבו התפקידים מתחלפים
Open Table Format	מפרט שמפריד מטא-דאטה מדאטה ומאפשר לכל מנוע לקרוא ולכתוב אותה טבלה
Apache Iceberg	פורמט טבלה פתוח — מפרט וסט ספריות, לא פלטפורמה
S3 Tables	bucket מסוג חדש: Iceberg מנוהל ב-data plane של S3, כולל compaction ותחזוקה
Materialized View	תצוגה מחושבת מראש שמצליבה טבלאות ומחזירה לצרכן את החיתוך שלו מיידית

מונח	משמעות כפי שהוצגה
Vector Embedding	ייצוג מספרי רב-מימדי של פריט מידע, שמאפשר חיפוש לפי קרבה במרחב
S3 Vectors	אחסון וקטורים בסקייל של מיליארדים-טריליונים, לייטנסי sub-second בעלות S3
Elevation	העלאת אינדקס מ-S3 Vectors ל-OpenSearch Serverless לפרק זמן שבו נדרש לייטנסי נמוך
Progressive Disclosure	קריאת פיסות מידע קטנות בזו אחר זו במקום טעינת כל ההקשר מראש
MCP	פרוטוקול אחיד לחיבור מודל לכלים ולמקורות מידע; SnowChat מומש כ-MCP server