מ-vibe coding לפרודקשן:
אפליקציות AI על Amazon Bedrock

tlv-aim201 — סיכום סשן, AWS Summit Tel Aviv 2026

Yoav Fishman, Sr. Solutions Architects, AWS | Dr. Yonatan Molad-Hayo, Director of- ו Itay Heshes
Research Platform, Aidoc

10 בספטמבר 2026 | משך: כ-43 דקות | הופק מהקלטת הסשן

מסלול: Building AI and Agentic Applications | רמה: 200

על המסמך הזה

המסמך מסכם את הסשן tlv-aim201 מתוך AWS Summit Tel Aviv 2026. הוא נבנה מתמלול אוטומטי של ההקלטה המלאה ומהסליידים שחולצו מהווידאו עצמו, כך שאפשר לחזור לתוכן בלי לצפות שוב ב-43 הדקות. חותמות הזמן בגוף המסמך מפנות לנקודה המדויקת בהקלטה.

הסשן בנוי משני חצאים שונים לגמרי באופיים. בחצי הראשון שני Solutions Architects מ-AWS מובילים סיפור מסגרת אחד — בוט לרשת מסעדות בשם ShefBot — ומשתמשים בו כדי לעבור, שירות אחרי שירות, על כל התחנות שבין דמו שנבנה ב-vibe coding לבין מערכת שרצה מול משתמשים אמיתיים. בחצי השני מגיע מקרה אמיתי: Aidoc מספרים איך הם מודדים ביצועים של מודלי הדמיה רפואית בסקייל, כשטעות היא לא באג אלא ממצא מסכן חיים שפוספס.

איך לקרוא את זה

- **פרק 1 — תקציר מנהלים:** חמש דקות קריאה. מה הוצג ומה שווה לקחת.
- **פרקים 2–5 — שכבת המודל והדאטה:** בחירת מודל, הערכה מבוססת דאטה, RAG ו-Knowledge Bases, ו-AWS Context.
- **פרקים 6–8 — השכבות שהופכות דמו למוצר:** מידע חיצוני בזמן אמת, AgentCore ,Guardrails ו-AIOps.
- **פרקים 9–13 — המקרה של Aidoc:** הבעיה הקלינית, ה-Analyzer, מנועי ההרצה, והמדידה שמאחורי אישורי ה-FDA.
- **פרק 14 — מה לוקחים מכאן:** מסקנות, ומילון מונחים בסוף.

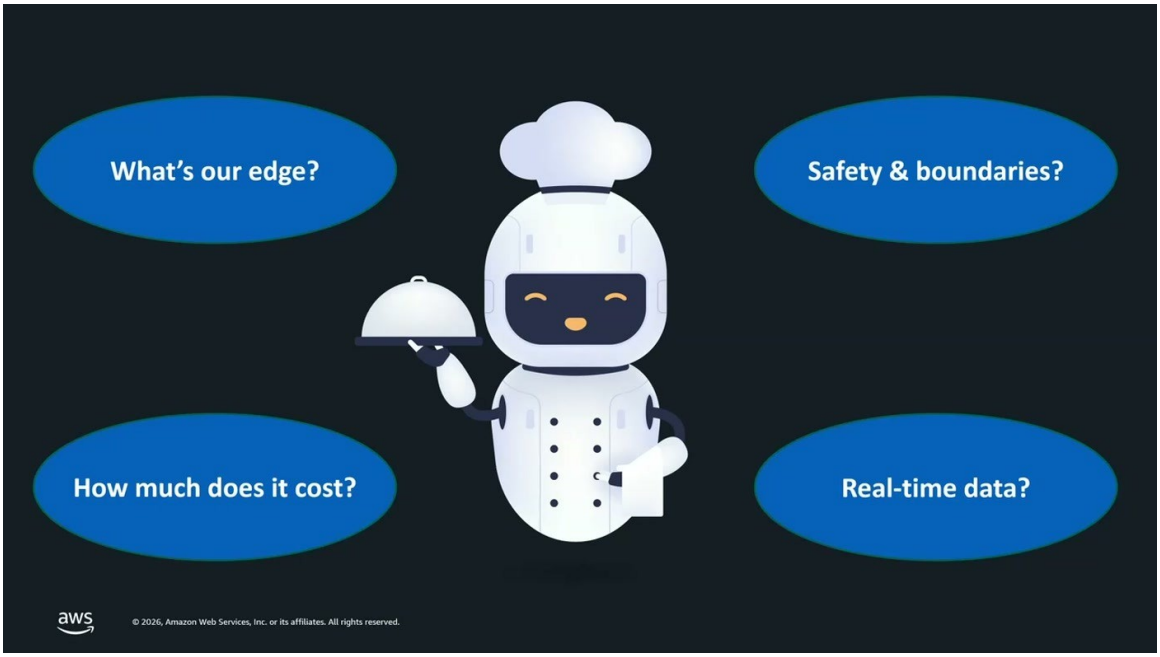
הערת מקור המסמך הופק מתמלול אוטומטי של ההקלטה. התמלול מדויק בקטעים העבריים, אך שמות מוצרים ומונחים לועזיים הופיעו בו בשיבוש פונטי (למשל "אג'נט קור" = AgentCore, "בדרו גארדרלס" = Bedrock Guardrails) — הם נורמלו לשם האנגלי מול הסליידים. הציטוטים מובאים כלשונם בתמלול ואומתו בחותמת הזמן המצוינת; במקומות שבהם ה-ASR שיבש מספר או שם מוצר, הדבר מצוין בגוף הטקסט. מספרים שמופיעים על הסליידים סומנו ככאלה.

1. תקציר מנהלים

הסשן לא מנסה לשכנע שאפשר לבנות אפליקציית AI. הוא יוצא מנקודת הנחה שכולם כבר בנו אחת — ושואל למה היא לא בפרודקשן. המצגת פותחת בשאלה לקהל: מי בנה chatbot עם vibe coding בשנה האחרונה, ומי העלה אותו לפרודקשן אחרי שבוע? אחרי חודש? הפער בין שתי השאלות הוא כל התוכן של 43 הדקות.

- **הפער בין דמו לפרודקשן הוא לא פער של מודל, אלא של מעטפת.** המצגים מפרקים אותו לשבע תחנות: בחירת מודל, הערכה, חיבור לדאטה ארגוני, קשרים בין הנתונים, מידע חיצוני, guardrails, ותפעול. אף אחת מהן לא נוגעת באיכות המודל עצמו.
- **מודל אחד לכל הפתרון הוא אנטי-פטרן.** ההמלצה החוזרת היא בחירת מודל ברמת המשימה, לא ברמת האפליקציה: מודל reasoning למשימה מורכבת, מודל קטן ומהיר להמרת יחידות מידה. איתי מסכם ב-[40:25]: "כדאי שנפתח את השריר הזה שמאפשר לנו להחליף בין מודלים במידת הצורך".
- **בחירת המודל צריכה להיות מבוססת מדידה, לא תחושה.** הוצגו שני כלים — Amazon Bedrock Evaluations (הערכה אנושית או אוטומטית מול ground truth) ו-Advanced Prompt Optimization, שמשכתב את ה-prompt ומשווה עד חמישה מודלים במקביל.
- **שני שירותים הוצגו כחדשים: AWS Context ו-Web Grounding & Web Search.** AWS Context (בסטטוס preview לפי הסלייד) בונה knowledge graph אוטומטי מעל הדאטה הארגוני ומנגיש אותו לסוכנים; שירות ה-Web Search מאנדקס מסמכים מהאינטרנט בתוך AWS כך שהסוכן מקבל מידע עדכני — "אין תקשורת החוצה, אין איגרס בכלל" [16:52].
- **Aidoc סיפקו את הצד השני של המשוואה: איך מוכיחים שהמודל עובד.** לפי הנתונים שהציגו הם מותקנים בכ-2,000 בתי חולים, עם יותר מ-40 פתרונות מאושרי FDA ושירות ליותר מ-60 מיליון מטופלים בשנה [26:05]. המערכת שהציגו לא מייצרת תחזיות — היא מודדת תחזיות.
- **התובנה המרכזית של Aidoc: הפער הוא הנדסי, לא מודלי.** "large language models הם באמת מהפכה, הם אינטליגנטיים מאוד, אבל אמון ובכלל אימפקט אמיתי בפרודקשן זאת המעטפת האנג'נירית שמסביב להם" [38:28].

2. נקודת הפתיחה: ShefBot וארבע השאלות



ארבע השאלות שמנהל המוצר מחזיר לצוות אחרי הדמו: מה היתרון שלנו, מה הגבולות, מה זה עולה, ומאיפה מגיע מידע בזמן אמת.

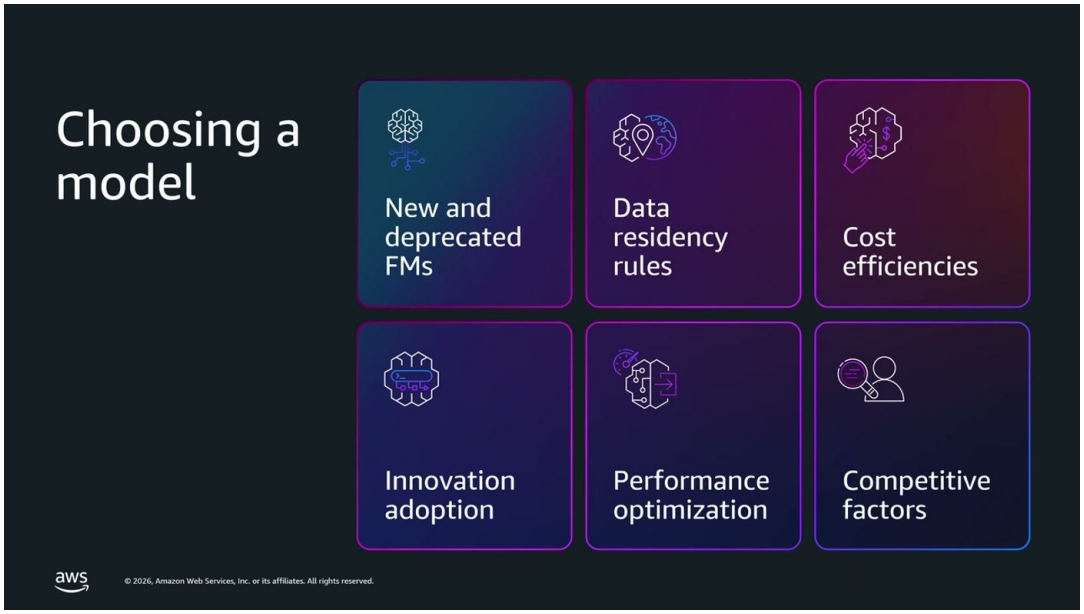
הסיפור נפתח בתרחיש מוכר: פרויקט חדש ב-Kiro (סביבת הפיתוח האג'נטית של AWS), תיאור של מה שרוצים, כמה שעות של איטרציות — ויש אפליקציה עובדת. בדוגמה זו ShefBot, בוט לרשת מסעדות בת שלושים סניפים שעונה על שאלות כמו "מה נמכר הכי הרבה בחודש האחרון".

הדמו עובד. ואז מגיעות השאלות. איתי מציג אותן כארבע בועות על המסך, ומודה שהן שאלות טובות: מה היתרון שלנו על פני ChatGPT? האם יש כאן בכלל נתונים בזמן אמת? מה קורה כשמישהו שואל שאלה שהבוט לא אמור לענות עליה? וכמה זה עולה?

הוא בוחר להתחיל דווקא באחרונה, כי היא זו שמייצרת את הלחץ: טוקנים שווים דולרים, וככל שהבוט מדבר יותר כך החשבון גדל — ובסקייל של שלושים סניפים ומאות שאלות ביום זה מצטבר.

> **Itay Heshes, [0:00] —** אז אנחנו מבינים שיש פער בין לבנות משהו מהיר עם vibe coding לבין להעביר אותו לפרודקשן, ויש לנו דרך לעבור.

3. בחירת מודל: שישה שיקולים, ולמה זו החלטה חוזרת

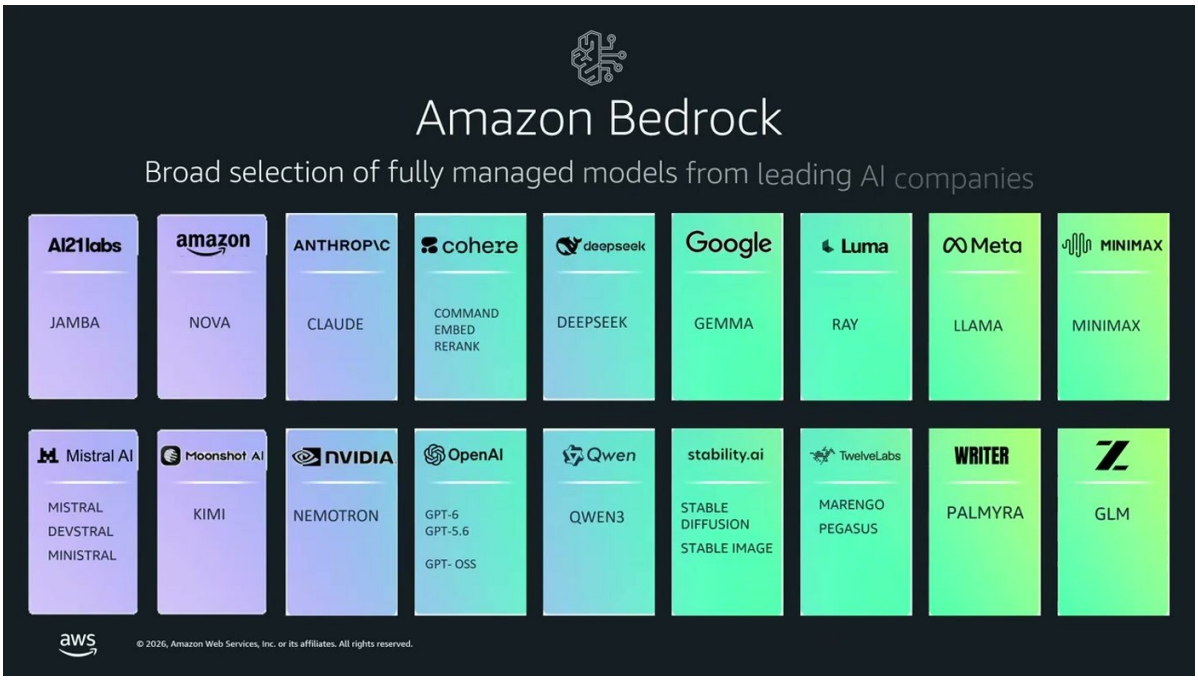


שישה שיקולים לבחירה ולהחלפה של מודל, כפי שהוצגו על הסלייד.

הנקודה המרכזית כאן היא שבחירת המודל אינה החלטה חד-פעמית בתחילת הפרויקט. כשמתחילים לבנות לוקחים בדרך כלל את אחד המודלים המתקדמים האחרונים כדי להגיע להוכחת היתכנות מהר; החלק הקשה מתחיל אחר כך.

הסלייד מונה שישה שיקולים: מודלים חדשים ומודלים מיושנים (new and deprecated FMs), כללי data residency, יעילות עלות, אימוץ חדשנות, אופטימיזציית ביצועים, ושיקולים תחרותיים. שניים מהם הודגשו בדיבור: מודלים ישנים מגיעים ל-deprecation ואז אין ברירה אלא להחליף; ואפליקציה שדורשת latency נמוך עשויה להסתפק במודל קטן ומהיר.

> **Itay Heshes, [3:03] —** מודלים חדשים יוצאים כל הזמן אנחנו רוצים להיות מסוגלים להחליף להתחיל להשתמש המודל החדש ברגע שהוא נגיש, בנוסף תזכרו שהמודלים הישנים הולכים ו... deprecated.



מפת ספקי המודלים הזמינים ב-Amazon Bedrock כפי שהופיעה על הסלייד — בהם AI21, Amazon, Anthropic, Cohere, DeepSeek, Google, Luma, Meta, MiniMax, Mistral, Moonshot, NVIDIA, OpenAI, Qwen, Stability, TwelveLabs, Writer ו-Z.ai.

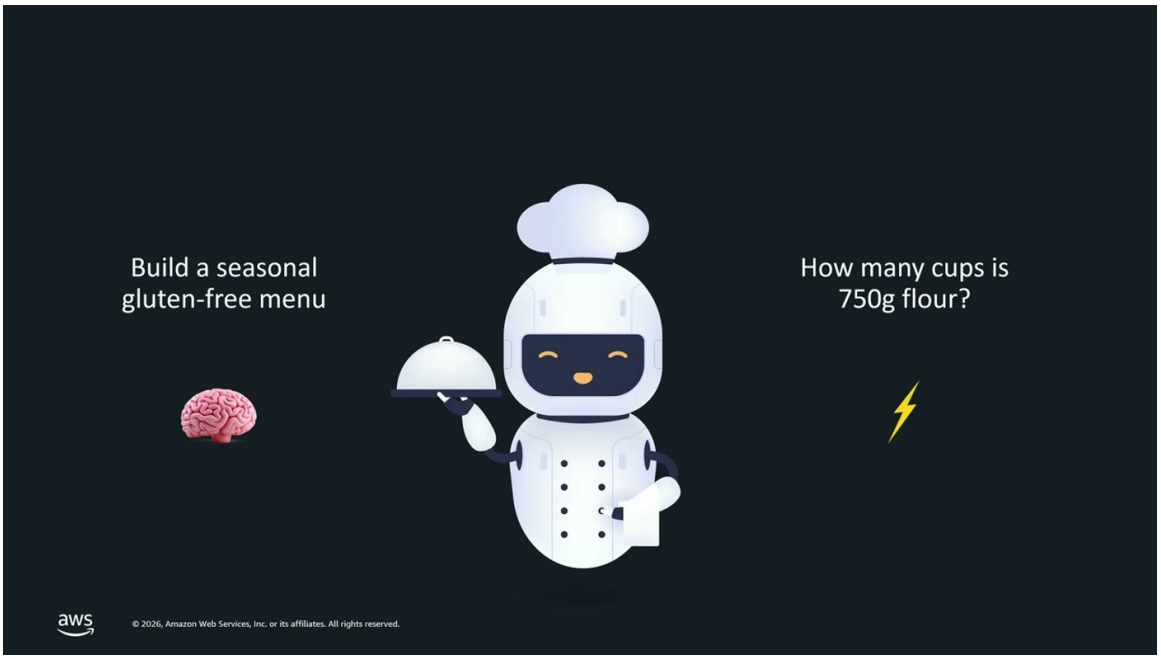
ההצגה של קטלוג המודלים נגעה בשלוש נקודות. ראשית, כל משפחת Claude של Anthropic זמינה ב-Bedrock. שנית, שיתוף הפעולה עם OpenAI הביא את מודלי ה-frontier שלהם ל-Bedrock — התמלול משבש את שמות הגרסאות, והדובר מציין שמודל נוסף עלה לפלטפורמה "ממש לפני יומיים" (השם בתמלול משובש ולכן לא צוטט כאן כמספר גרסה ודאי). שלישית, עשרות מודלי open weights כמו Qwen, DeepSeek ו-Gemma.

Itay Heshes, [4:36] — אמזון בדרוק מנגיש לנו מבחר גדול של מודלים בגדלים שונים עם עלויות ויכולות משתנות.

היתרון התפעולי שהודגש: תשלום לפי צריכת טוקנים בפועל, סקלאביליות ותמיכה כמו כל שירות AWS אחר — ומעטפת האבטחה והקומפליינס. הסלייד הבא הראה שתי מילים בלבד, Secure ו-Compliant, והדובר פירט מה מאחוריהן.

Itay Heshes, [6:10] — מה שזה אומר שהמידע שלכם נשאר שלכם, הוא מוצפן תמיד, הוא לא זמין לאותם ספקיות המודלים ובטח לא משמש אותם לאימון.

4. מודל לכל משימה, ואיך בוחרים אותו מבלי לנחש



שתי משימות באותה אפליקציה: בניית תפריט עונתי ללא גלוטן מול המרה של 750 גרם קמח לכוסות — ולכל אחת מהן מתאים מודל אחר.

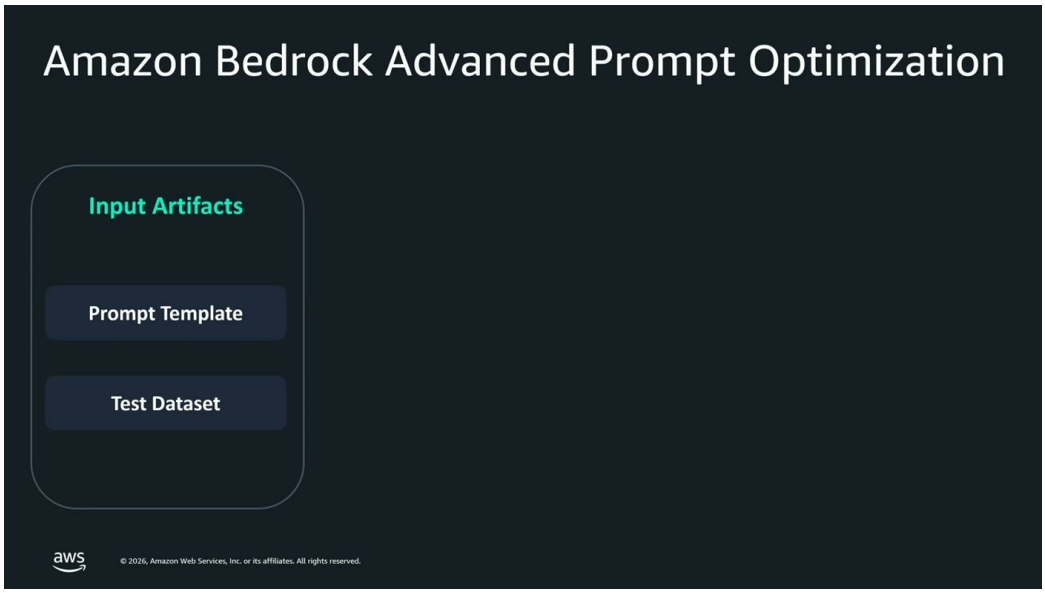
זו נקודת המפתח של החצי הראשון. השאלה "איזה מודל נבחר לפתרון" מוצגת כשאלה שגויה מיסודה. באותה אפליקציה עצמה יש משימות בעלות אופי שונה לחלוטין: בניית תפריט עונתי ללא גלוטן היא משימה שדורשת מודל מתוחכם עם יכולות reasoning; המרת יחידות מידה תקבל מענה מספק ממודל קטן, מהיר וזול בהרבה. ההחלטה צריכה להיות ברמת המשימה.

ומכאן השאלה השנייה — לפי מה בוחרים. התשובה של הסשן ברורה, וחוזרת גם אצל Aidoc בהמשך.

> **Itay Heshes, [6:10] —** אנחנו רוצים לעשות את זה על סמך דאטה, ולא על סמך תחושת בטן.

Amazon Bedrock Evaluations

- **Human Evaluation:** הכלי מספק ממשק שבו מומחה תוכן או לקוח עסקי מדרג ידנית את התשובות שהתקבלו מהמודלים השונים. התוצר הוא דוח השוואה בין המודלים לפי אותם דירוגים.
- **Automatic Evaluation:** בקלט מספקים prompt dataset — השאלות שרוצים לבחון — ואפשר לצרף גם ground truth והגדרת מטריקות. יש מטריקות מובנות בכלי, ואפשר להגדיר custom metrics. ה-job מריץ את השאלות על המודלים שנבחרו ומחזיר evaluation report לפי אותן מטריקות.



ה-Input Artifacts של Advanced Prompt Optimization: תבנית prompt ו-test dataset.

הכלי השני, Amazon Bedrock Advanced Prompt Optimization, עושה שני דברים בבת אחת — אופטימיזציה של ה-prompt והשוואה בין מודלים. מזינים לו prompt template ו-dataset שמשמש ground truth; הוא מריץ inference מול המודלים השונים, מבצע evaluation מול ה-ground truth, ובמידת הצורך משכתב את ה-prompt. הכול אוטומטית, ולפי הדובר עד חמישה מודלים במקביל.

> **Itay Heshes, [7:40] —** התוצר שנקבל הוא גם פרומט אופטימלי שמותאם לאותם מודלים וגם מדדי איכות כמו עלות ולטנסי.

5. הדאטה הארגוני: מ-RAG ל-Managed Knowledge Bases

אחרי שבחרו מודל והורידו עלויות, חוזרים ל-ShefBot ושואלים מה המנה הנמכרת ביותר בשבוע האחרון. התשובה שמתקבלת היא גנרית לחלוטין — הדגמה נקייה של הבעיה.

> **Itay Heshes, [9:17] —** פסטה פרימוורה, איזה פופולר דיש. תשובה גנרית, אותה תשובה שתקבל כל מסעדה בעולם.

המידע שדרוש לתשובה אמיתית כבר קיים בארגון: דוחות מכירה, נתוני inventory ומלאי חסר, פידבקים מלקוחות. השאלה היא איך מגישים אותו למודל. הסלייד מציג את מנעד האפשרויות כסולם עולה — prompt engineering, ואז RAG, ואז customization, ועד אימון מודל מאפס — כשהציר האנכי הוא Complexity, Cost, Time. ככל שמתקדמים ימינה, שלושתם עולים. עבור ShefBot בחרו RAG.

ארכיטקטורת ה-RAG הוצגה ברמת eye-level בלבד, עם שני חלקים מרכזיים: Data Ingestion — לקיחת הדאטה הארגוני ואינדוקס שלו למסד וקטורי, על כל הטכניקות והבחירות שנלוות לכך; ו-Text Generation — שליפת המידע הרלוונטי בזמן השאלה והזרמתו למודל. הפואנטה היא שכל זה דורש עבודת תשתית ו-moving parts רבים, וזו הסיבה לקיומו של Amazon Bedrock Knowledge Bases.

> **Itay Heshes, [10:47] —** אתם מספקים את המידע, בוחרים את המימוש ובעזרת קונפיגורציה פשוטה ומקבלים API לחיפוש סמנטי וגם למענה על שאלות.

שש היכולות של Amazon Bedrock Managed Knowledge Bases כפי שנמנו על הסלייד.

ההכרזה החדשה שהוצגה היא Amazon Bedrock Managed Knowledge Bases — צעד נוסף בפישוט העבודה מול המידע הארגוני. מה שנמנה עליה: קונקטורים ל-S3, SharePoint, Google Drive ואחרים; managed store; smart parsing; אינטגרציה עם AgentCore; ומוכנות לפרודקשן. היא מוצגת כבנויה "day one" לעבודה עם agents.

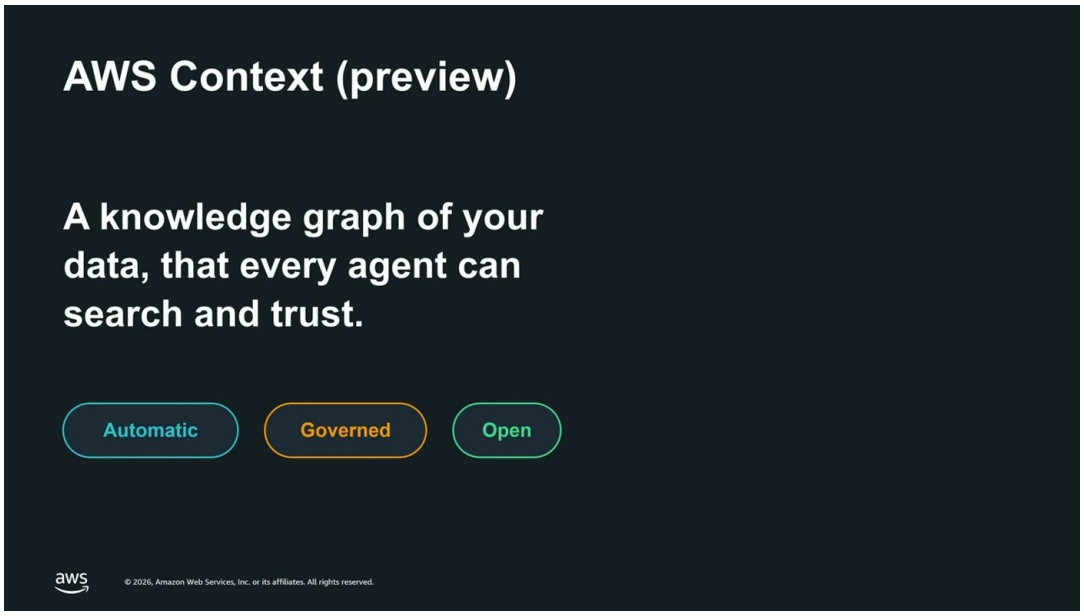
היכולת שקיבלה את ההסבר הארוך ביותר היא Agentic Retrieval: היא מאפשרת לקחת שאלה מורכבת ולפצל אותה לשאלות פשוטות יותר. בדוגמה שניתנה — "מה נמכר הכי הרבה בשבוע האחרון בסניף הכי רווחי" — המערכת קודם מוצאת מהו הסניף הרווחי, ורק אז מחפשת בו את המנה. התשובה המלאה מוחזרת למשתמש.

6. AWS Context: הקשרים בין פיסות המידע

כאן יואב פישמן עולה לבמה וממשיך את אותו סיפור, עם תרחיש חדש: ספק הסלמון מתקשר ומודיע שבעוד יומיים לא תהיה אספקה, בגלל תקלה במשאיות הקירור. השאלה שרוצים לשאול את המערכת היא מורכבת מדרגה: אילו סניפים ייפגעו, אילו מנות ייפגעו, ומאילו ספקים באזור אפשר להשיג חלופה.

כל המידע הדרוש קיים — ספקים, חברות שליחויות, לוקיישנים, תפריטים, מנות. הבעיה היא אחרת.

> **Yoav Fishman, [13:48] —** המידע הוא מפוזר. הוא נמצא רחוק אחד מהשני. מה שחסר לי כדי שאני באמת אוכל להוציא ממנו תובנות ולענות על שאלות מורכבות, זה קשרים חכמים.



שלוש המילים שמגדירות את השירות על הסלייד: Automatic, Governed, Open — וסטטוס preview לצד השם.

AWS Context הוצג כשירות חדש ובסטטוס preview. הוא יושב על הדאטה — structured ו-unstructured כאחד — ובונה ממנו knowledge graph באופן אוטומטי, שמחבר את פיסות המידע זו לזו ומנגיש אותן לסוכנים.

> **Yoav Fishman, [13:48] —** הוא יושב על המידע שלכם גדול ככל שיהיה, structured, unstructured, ובעצם יוצר knowledge graph בצורה אוטומטית, מחבר את כל הפיסות המידע בצורה חכמה.

שכבות הארכיטקטורה כפי שהוצגו

שכבה	מה יש בה
Metadata	מידע על המידע — הקשרים החכמים שמחברים בין ישויות הדאטה. נשמר ב-Apache Iceberg, בפורמט פתוח.
Knowledge Graph	נבנה אוטומטית מעל שכבת המטא-דאטה, ומשתפר ומתחזק ככל שעושים בו יותר שימוש.
Governance	עוטף את הגרף כך שכל משתמש רואה רק את המידע שההרשאות שלו מתירות.
Agentic Search API / MCP	החשיפה החוצה — API, או חיבור של הסוכנים כ-tool דרך MCP.

הערה שחשוב לשים לב אליה: אף שהגרף נבנה אוטומטית, יואב מדגיש שיש מקום למומחה הדאטה של הארגון להיכנס ולהעשיר אותו — להוסיף קשרים ולהטמיע business logic. זו לא מערכת שמחליפה את מודל הנתונים אלא כזו שמאיצה את בנייתו.

7. מידע שאינו שלכם: Web Grounding & Web Search

התחנה הבאה היא מידע חיצוני. ShefBot אחראי גם על רכש, והרכש תלוי במזג האוויר: בגל חום נמכרת יותר גלידה ופחות מרק, והמרכיבים טריים — כלומר ההזמנה צריכה להשתנות מהר. מזג האוויר הוא לא דאטה פרופריטרי, והוא משתנה כל הזמן.

החלופה הרגילה היא לעבוד מול third-party services, לנהל API keys חיצוניים, ולדאוג ל-security, operations וסקייל של החיבור הזה. השירות שהוצג מחליף את כל השרשרת הזו בחיבור בודד ברמת tool.

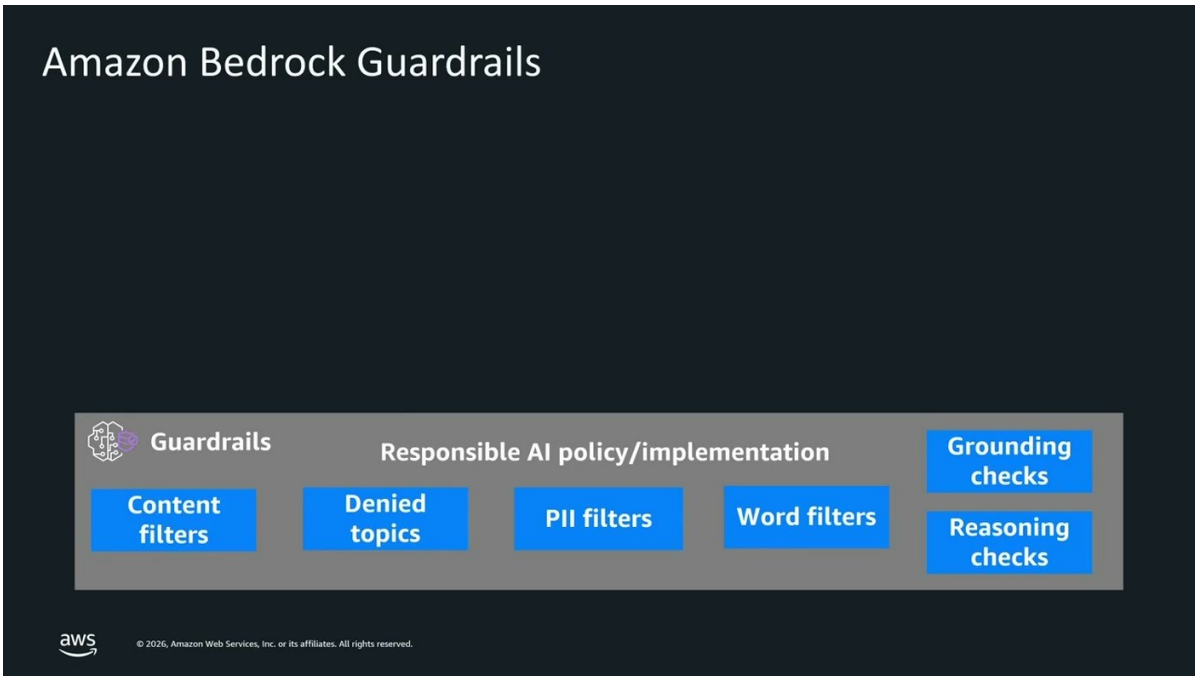
> **Yoav Fishman, [16:52] —** אנחנו לוקחים עשרות מיליארדי מסמכים מרחבי האינטרנט, מאתרים גדולים ואתרים קטנים כאחד, ומאנדקסים אותם אלינו.

- **רענון תכוף:** לפי הדובר, הדאטה מתרענן "אחת לכל כמה דקות", כדי שה-search של הסוכן יחזיר את המידע הרלוונטי לאותה נקודת זמן.

- **ללא יציאה החוצה:** "אין תקשורת החוצה, אין איגרס בכלל, כל השאילתות שלכם נשארות בפנים, הכל על הרשת הפנימית של AWS ועל התשתית של AWS" [16:52].
- **latency מותאם real-time:** הוצג כדרישה מפורשת לאפליקציות שמשרתות משתמש חי.

8. Guardrails: מה שקורה כשמשתמשים אמיתיים נכנסים

ברגע שמפגישים את המערכת עם משתמשים אמיתיים, אין שליטה על מה שנשאל. הדוגמה שהוצגה על המסך היא שאלה ש-ShefBot בהחלט לא נבנה בשבילה: "עבור מי אני צריך להצביע בבחירות" [18:22]. מעבר לכך שזו לא מטרת המערכת, יואב מציין שזה גם מייצר עלויות טוקנים מיותרות.



סוגי הבקרות ב-Guardrails: content filters, denied topics, PII filters, word filters, ולצדם grounding checks ו-reasoning checks.

מה שנמנה בדיבור כמה שאפשר להגדיר: חסימת תכנים פוגעניים ואלימים, prompt injections, מניעת דיון בנושאים מסוימים (פוליטיקה, פיננסים, משפט), ואפילו מניעת שימוש במערכת ככלי לכתיבת קוד — שימוש שרק מנפח את החשבון ולא לשם כך היא נועדה. בנוסף: מיסוך מידע רגיש כמו מספרי טלפון ותעודות זהות, וחסימת מילים ספציפיות.

ההסבר הטכני נסוב על מיקום ה-Guardrails ב-flow: ה-prompt לא מגיע ישירות למודל אלא פוגש קודם את הפוליסה, ורק אם הוא עובר אותה הוא ממשיך. אותו דבר בכיוון ההפוך — התשובה שהמודל ייצר נבדקת מול הפוליסה לפני שהיא ממשיכה ל-downstream services, לסוכנים או למשתמשים. שתי נקודות בדיקה, לא אחת.

> **Yoav Fishman, [19:54] —** השירות הזה יעבוד עם כל מודל מכל מודל פרווייידר שתחליטו לעבוד איתו וגם עם מודלים שאתם עושים להם hosting באינפרסטרקטשר שלכם, במידה וזה קורה על EC2 או על pod בקלאסטר קוברנטיס.

> **שווה לשים לב** זו הנקודה הפרקטית ביותר בחצי הראשון של הסשן: Guardrails אינו כבול למודלים שרצים ב-Bedrock. הוא נצרך כ-API נפרד ב-serverless ו-pay-as-you-go, ולכן אפשר להחיל אותו גם על מודל עצמאי שמתארח בתוך הארגון.

9. AgentCore ו-AIOps: מה שמתחיל אחרי ההעלאה

אם באפליקציה יש רכיבים אג'נטיים, נדרש טיפול שמערכות קלאסיות לא הצריכו. הרשימה שנמנתה: ניהול זהות של הסוכן, בשם מי הוא פועל ומה ההרשאות שלו ושל המשתמש; session isolation בין ריצה לריצה; וניהול זיכרון לטווח קצר ולטווח ארוך, כדי שהפעולות יסתמכו על מה שכבר נעשה ולא יתחילו כל פעם מדף חלק. Amazon Bedrock AgentCore הוצג כאוסף אבני הבניין שמכסות את הדברים האלה. יואב הפנה במפורש לסשן שאחריו באותו אולם לצלילה לעומק בנושא.

מכאן עוברים ל-AIOps — החלק שהמצגים מדגישים שהוא לא סוף הדרך אלא תחילתה.

> **Yoav Fishman, [21:30] —** ברגע שאנחנו עולים לפרודקשן לא רק שלא סיימנו את העבודה, עכשיו יש עוד דברים שאנחנו צריכים לדאוג להם.

- **Observability:** לוודא שהמערכת מתנהגת כפי שתוכננה, שה-latency בגדר שהוגדר, ושהעלות לא משתוללת — "בטח ובטח שמתעסקים עם agentים וטוקנים שהקוסט יכול להיות יקר מאוד מהר" [21:30].
- **Evaluation מתמשך:** כשמוסיפים פיצ'רים, מחליפים מודל גדול בקטן או משנים prompts — צריך שכבה שתאמת שהשינוי אכן מטיב עם המערכת ולא עושה בדיוק את ההפך.

10. Aidoc: כשטעות היא לא באג

החצי השני של הסשן הוא סיפור מקרה. ד"ר יונתן מולד-חיו, דירקטור פיתוח ב-Aidoc שמוביל את קבוצת Research Platform, מציג מערכת פנימית בשם Imaging AI Evaluation Platform. הוא פותח בתרגיל דמיון: קרוב משפחה מגיע לחדר מיון עם כאב חדש ולא מוסבר בראש או בחזה, ונשלח להדמיה — CT או X-ray.

שלוש הבעיות שהוצגו

הבעיה	מה היא אומרת בפועל
מיון ותיעדוף	ההדמיה נכנסת לתור המשימות של הרדיולוג התורן, שעומד מול עשרות הדמיות לפענוח. היא עלולה להיכנס במקום השלושים ברשימה, וההמתנה נמשכת שעות.
טעויות באבחון	לפי הנתון שהוצג, בעד 20% מהמקרים — כולל מצבים מסכני חיים — הרדיולוג עלול לקבוע שמדובר בבעיה אחרת ממה שיש למטופל בפועל.
עיכובים בטיפול	גם כשהזיהוי נכון, ההחלטה והטיפול דורשים צוות מולטי-דיסציפלינרי. אם זה נדרש בשלוש לפנות בוקר והמנתח או המצנתר ישן בבית, נוצר עיכוב משמעותי.

מסלול העבודה של פלטפורמת ה-AI הקלינית של Aidoc, מסריקת המטופל ועד שילוב בתהליכי העבודה הקליניים.

הפלטפורמה הקלינית שפיתחו במשך קרוב לעשור מנתחת הדמיות רפואיות, מתאימה לכל הדמיה את המודל או המודלים המתאימים, ומחזירה את התוצאה למערכות בית החולים — שם היא משמשת לתיעדוף רשימת העבודה ולהנעת תהליכי טיפול קליניים.

> **Dr. Yonatan Molad-Hayo, [26:05] —** אנחנו מותקנים בכאלפיים בתי חולים בכל רחבי העולם ופיתחנו עד היום יותר מ-40 פתרונות מאושרי FDA, מה שמאפשר לנו לתת שירות ליותר מ-60 מיליון מטופלים בשנה.

ומיד אחרי המספרים האלה מגיעה הטענה שמסבירה למה בכלל קיימת פלטפורמת ההערכה.

> **Dr. Yonatan Molad-Hayo, [26:05] —** מודל חכם וחזק ככל שיהיה אין לו ערך אמיתי אם אנחנו לא יכולים להוכיח בצורה טובה ומדויקת שהוא עומד במדדי ביצוע ואיכות גבוהים מאוד.

11. מתי מודדים, ואיזו ראיה משתמשת כ-ground truth

ההערכה מתרחשת בשלושה מועדים שונים, ולכל אחד תפקיד אחר. בשלב המחקר והפיתוח — כדי לדעת איפה המודל עומד בין "בהתהוות" ל"מוכן להפצה". בהפצה לבית חולים חדש — כל בית חולים הוא סביבת הרצה ייחודית,

עם מערך דמוגרפי שונה, סוגי מכשירי הדמיה שונים ופרוטוקולי סריקה שונים, והחובה היא לוודא שהביצועים בשטח טובים לפחות כמו במעבדה. ובאופן שוטף — בכל בית חולים שבו הם מותקנים, כדי לזהות שינויים בלתי צפויים באיכות הביצועים, כשהאתגר הוא לזהות אותם בזמן.

השיטה הראשונה הייתה להיעזר ברדיולוגים שיפענחו סריקות ללא הטיה ולהשוות לתוצאות המודל. בסקייל של עשרות מיליוני סריקות זה לא מחזיק מים. הפתרון שבחרו מתבסס על משהו שממילא קורה בבית החולים: התיעוד הקליני הטקסטואלי שהרדיולוג כותב זמן קצר אחרי הסריקה. כלומר שני סיגנלים — תוצר ה-Imaging AI מול הדוח הרדיולוגי הכתוב — והצלבה שלהם בקוהורט גדול מייצרת את שכבת הדאטה שממנה מחשבים ביצועים.

כותרת האתגר הראשון כפי שהופיעה על הסלייד: דוחות קליניים אינם תוויות.

וכאן האתגר הראשון, והוא לא ייחודי לרפואה: הדוח הרדיולוגי אינו תווית. הוא טקסט חופשי, מקצועי, שנכתב על ידי רופאים עבור רופאים, ולעולם אינו מסתכם בתוצר סכמטי נוח לעיבוד.

> **Dr. Yonatan Molad-Hayo, [29:09] —** הסגנון והשפה משתנים בין רדיאולוגים שונים, בתי חולים וכמובן גם ארצות, והוא אף פעם למעשה לא מסתכם לאיזה שהוא תוצר סכמטי שנוח לעבוד איתו להמשך עיבוד תוכנתי.

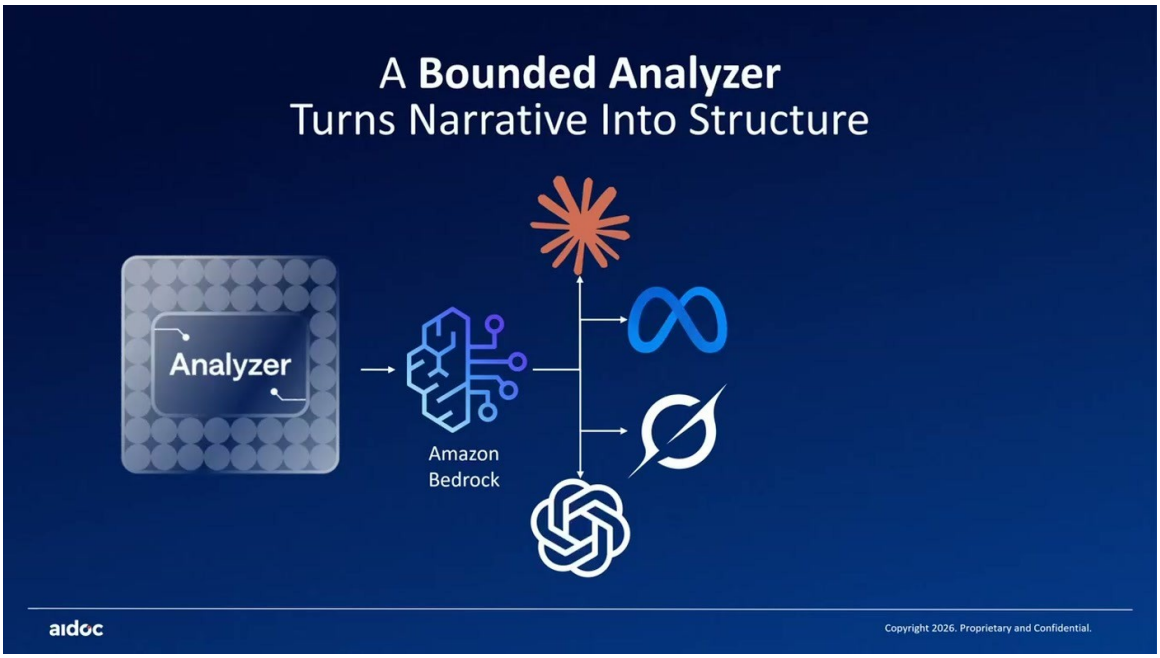
מה שמסתתר בתוך דוח כזה, לפי הפירוט שניתן: משפטי שלילה, משפטי חוסר ודאות, נתונים היסטוריים על המטופל, והתייחסות להדמיות קודמות — כלומר ייתכן משפט שמתייחס לסריקה אחרת לגמרי מזו שעליה נכתב הדוח. וכל זה לצד הממצאים עצמם, על הכמות, הגודל, המיקום האנטומי המדויק, ולעתים גם ציון האם הייתה התערבות רפואית מוקדמת באותו איבר.

הדוגמה שהוצגה על המסך היא משפט בודד שחולץ מדוח ארוך, וממנו נגזרים כל הפרמטרים לחישוב: סוג ההדמיה (CT אנגיוגרפי), סוג הממצא (אניוריזמה סקולרית), הגודל (ארבעה מילימטרים), המקטע והעורק המדויקים — ובאותו משפט גם שלילה מפורשת של סימני התערבות רפואית מוקדמת.

12. ה-Analyzer: למה לא פשוט לשלוח את הדוח ל-LLM

המחשבה הנאיבית, כלשונו, היא לשלוח את הדוח כולו לאחד ממודלי ה-LLM החזקים, אולי עם הנחיה למבנה התוצר הרצוי. הוא שולל את זה מפורשות: "גם אם זה היה נכון, ואני יכול להגיד לכם שזה לא, בסטינג רפואי קליני שלנו זה לא באמת פרקטי" [32:15].

הנימוק הוא שבחירת המודל היא פעולה לא טריוויאלית שמשקללת בבת אחת latency, קונסיסטנטיות של התשובות, שיקולי אבטחה, קומפליינס, privacy, דיוק ועלויות — והשאלה אצלם מתחדדת עוד שלב אחורה: איזה מודל נכון להריץ באיזה שלב, עבור כל פתולוגיה בנפרד.



ה-Analyzer פונה דרך Amazon Bedrock לספקי מודלים שונים — שכבת ביניים אחת מול כל הספקים.

ה-Analyzer הוא יחידת הבסיס של המערכת: רכיב שלוקח טקסט רפואי מורכב והופך אותו לתוצר סכמטי מוגדר עבור פתולוגיה ספציפית. הוא מתואר כ"מומחה תוכן" צר — לא מודל כללי, אלא עץ החלטות.

> **Dr. Yonatan Molad-Hayo, [32:15] —** מדובר בעצם בקומפוננטה שממממשת עץ החלטות ייחודי וחכם, שמשלב שלושה סוגי מודלי NLP שונים.

- **מערכת חוקים דטרמיניסטית:** השכבה הזולה והצפויה ביותר, שמטפלת במה שאפשר להכריע בלי מודל.
- **מודלי NLP ייחודיים של החברה:** חלקם פותחו עוד לפני עידן ה-LLM, ועדיין משרתים חלק מהמסלולים.
- **מודלי LLM חזקים:** נצרכים דרך Amazon Bedrock, ומופעלים רק היכן שנדרש.

> **Dr. Yonatan Molad-Hayo, [32:15] —** והמטרה שלנו היא לעשות Model Chaining Dynamic, כלומר כל מודל נבחר על בסיס תוצאת המודל הקודם, והמטרה היא לייעל את הדיוק, להקטין את ה-latency וכמובן לאופטם את העלויות.

בקצה יושבת קומפוננטה שמבצעת validation ונורמליזציה, כך שלא משנה איזה טקסט נכנס ואיזה מסלול עבר — התוצר בסוף הוא סכמטי ואחיד. זו בדיוק הנקודה שאיתי העלה בחצי הראשון, אלא שכאן היא ממומשת ברמת הארכיטקטורה: הבחירה בין מודלים אינה קונפיגורציה סטטית אלא החלטה בזמן ריצה.

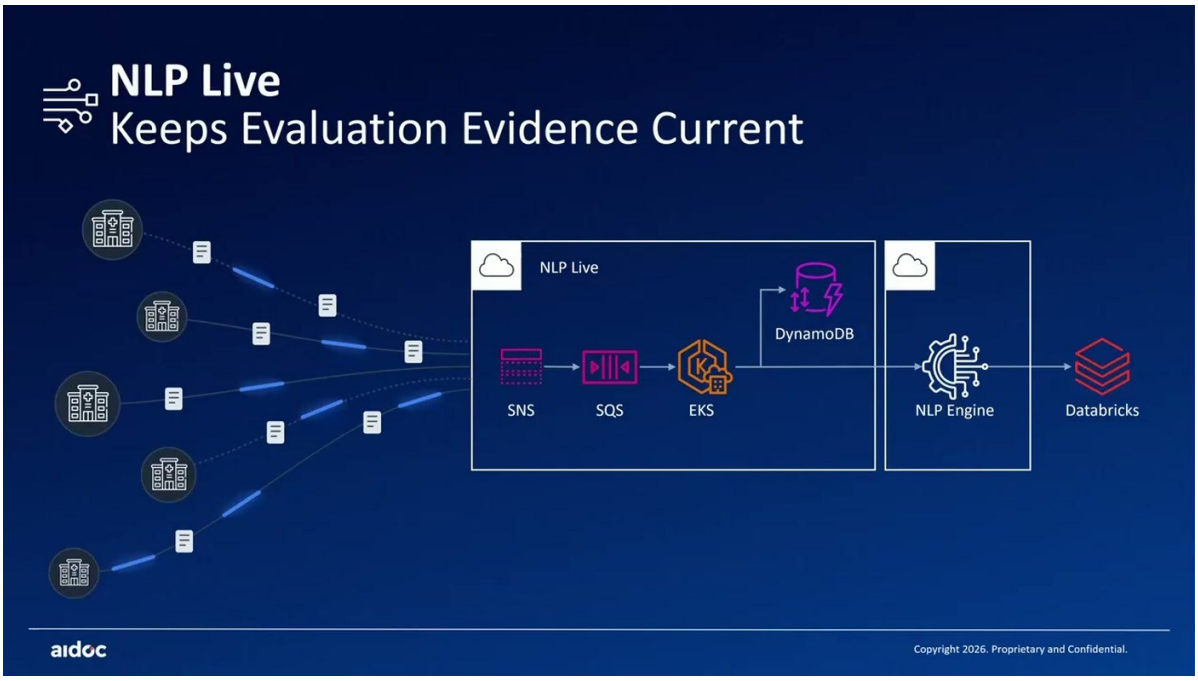
13. הרצה בסקייל: NLP Engine ו-NLP Live

אם יש יותר מ-40 פתרונות קליניים, יש גם עשרות Analyzers — לכל אחד עץ החלטות פנימי משלו ודרישות משאבי הרצה שונות. הסלייד מנסח את זה חד: One Analyzer Is Easy. Growth Requires a Platform. מכאן נולד ה-NLP Engine, שכבת האורקסטרציה העוטפת.

> **Dr. Yonatan Molad-Hayo, [33:46] —** זו מערכת מבוססת קוברנטיס, שמריצה את האנליזרים שלנו בסקייל, לפי דרישה מהלקוחות השונים, שיכולים להיות אגב חוקרים בתוך החברה או סרביסים מערכות אוטומטיות.

מה שהסנטרליזציה קונה, לפי ההסבר: התמודדות עם הרצות מקביליות רבות וסקייל גבוה, ובנוסף מנגנונים חוצי-מערכת כמו caching, retries, ושליטה ובקרה על ההרצות, על השימושיות ועל העלויות. התוצאות נשמרות ב-Data Warehouse של החברה.

> **Dr. Yonatan Molad-Hayo, [33:46] —** התבססנו על כל השירותים היפים האלה של AWS ובפרט Amazon Bedrock כקומפוננטת בסיס שדרכה אנחנו מוציאים את כל הפרומפטים שלנו לפרווידרים השונים, ובעצם קיבלנו כמעט את כל הדרישות שלנו out of the box — וזה יתרון אדיר עבור חברה כמונו.



הזרוע ה-event-driven: הדוחות מגיעים מבתי החולים דרך SNS ו-SQS ל-EKS, נרשמים ב-DynamoDB, עוברים ב-NLP Engine ומגיעים ל-Databricks.

ה-NLP Engine פותר את ההרצה לפי דרישה, אבל המדידה השוטפת דורשת זרוע שנייה — NLP Live. היא קולטת בשוטף את הדוחות הטקסטואליים מכל הלקוחות ב-real time, מזהה איזה Analyzer או Analyzers צריך להריץ על כל דוח, מפעילה אותם דרך ה-NLP Engine, ושומרת את התוצאות.

> **Dr. Yonatan Molad-Hayo, [35:23] —** זאת למעשה מערכת מבוססת אירועים, Event Driven, שקולטת בשוטף את הדוחות הטקסטואליים מכל הלקוחות השונים שלנו, Real Time.

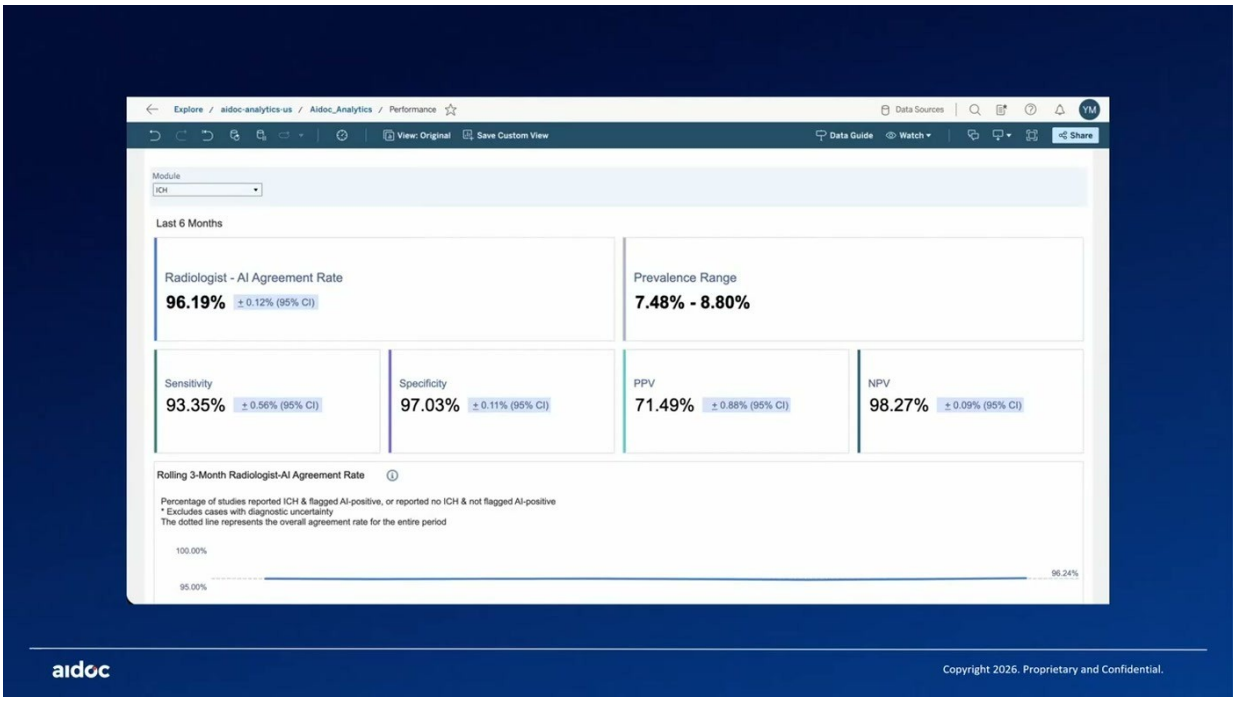
התוצר של שתי הזרועות יחד הוא מה שהוא מכנה "דאטאבייס חי ונושם של דוחות רדיולוגיים מאונדקסים" — כל דוח, על כל הפרמטרים שחולצו ממנו, צמוד לתוצאת מודל ה-Imaging AI לאותה סריקה. זהו מצע הדאטה שעליו רצים החישובים הסטטיסטיים.

14. מה סופרים, ולמי מראים את התוצאה

המדדים הם המקובלים במחקר: sensitivity, specificity, positive predictive value ועוד. אבל ההסבר שניתן להם אינו סטטיסטי אלא קליני, ומתורגם לשני תרחישים.

- **False positive — התראת שווא:** מתריעים לרופא על ממצא מסכן חיים שלא קיים בסריקה. מעבר לטרחה, "אנחנו גם עלולים לאבד את האמון והקרדביליות שהרופא ייתן במוצר שלנו" [36:53].
- **False negative — פספוס:** "דמיינו לכם שאנחנו לא נתריע לו על ממצא אקוטי מסכן חיים שכן הופיע בסריקה, זה מה שנקרא false negative, ובעברית אולי פספוס, ואתם יכולים להבין את ההשלכות לבד" [36:53].

הוא מוסיף ששקידה על המדידה המדויקת היא גם מה שסולל את הדרך לאישורי FDA למודלים. ואז מגיעה הנקודה שבה המדידה הופכת ממנגנון פנימי למוצר.

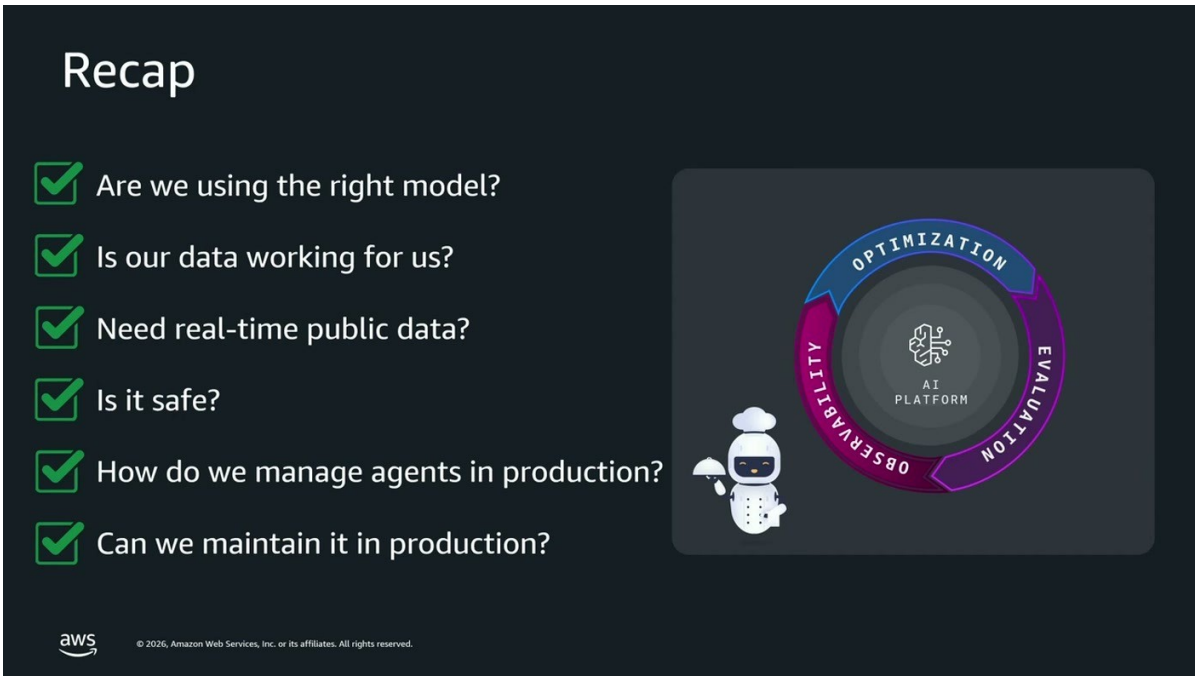


מסך מ-Aidoc Analytics עבור מודול ICH: Radiologist–AI Agreement Rate של 96.19%, Sensitivity 93.35%, Specificity 97.03%, PPV 71.49%, NPV 98.27% — כולם עם רווחי סמך של 95% (נתוני הסלייד).

Dr. Yonatan Molad-Hayo, [36:53] — אנחנו גאים להיות מבין החברות הבודדות בתחום שמשתפות בשקיפות מלאה עם הלקוחות שלה את הביצועים.

Aidoc Analytics הוא מערך דשבורדים שנפתח ללקוחות ומציג להם את ביצועי המודלים אצלם בשטח — כל בית חולים רואה את הנתונים שלו בלבד. הרציונל שהוצג הוא לולאה סגורה: שיקוף התוצאות מגדיל את האמון והביטחון של הרופאים במוצר, ואמון מגדיל את הסיכוי שהם ייעזרו בתוצאות בקבלת ההחלטות שלהם — וזה, בסופו של דבר, מה שמשפר את הטיפול במטופל.

15. מה לוקחים מכאן



סלייד ה-Recap שסגר את הסשן — שש השאלות שאיתי עבר עליהן, סביב מעגל של Observability, Evaluation ו-Optimization.

איתי חזר לבמה וסגר בשש שאלות שעל כל אחת מהן הסשן נתן כלי: האם אנחנו משתמשים במודל הנכון? האם הדאטה שלנו עובד בשבילנו? האם צריך מידע ציבורי בזמן אמת? האם זה בטוח? איך מנהלים agents בפרודקשן? והאם אפשר לתחזק את זה בפרודקשן? הוא הדגיש שלא בכל פתרון יידרשו כל הנקודות — אבל חשוב לשאול את כולן.

- **החליטו על מודל ברמת המשימה, ובנו את היכולת להחליף.** "כדאי שנפתח את השריר הזה שמאפשר לנו להחליף בין מודלים במידת הצורך" [40:25]. Aidoc לקחו את זה צעד קדימה ובוחרים מודל דינמית בזמן ריצה.
- **היתרון התחרותי הוא הדאטה, וגם הקשרים בתוכו.** "היתרון הכי גדול שיש לנו, הוא בעצם הדאטה שלנו. בואו נוודא שאנחנו משתמשים בו" [40:25] — ולא רק במידע עצמו אלא גם בקשרים בין פיסות המידע, כפי שהוצג ב-AWS Context.
- **Guardrails הוא לא פיצ'ר אופציונלי אלא תנאי לחשיפה למשתמשים.** "נוודא ששמנו גארדרלס כדי שהמוצר שלנו יענה רק למטרה שלשמה בנינו אותו" [40:25]. והוא עובד גם על מודלים שמתארחים מחוץ ל-Bedrock.
- **רכיבים אג'נטיים מוסיפים מורכבות ייעודית.** ניהול זיכרון, ניהול הרשאות, session isolation ו-observability — נושאים שהופנו לסשן העוקב במסלול.
- **AIOps הוא הדבר היחיד שאי אפשר לוותר עליו.** "אנחנו רוצים שהפתרון שלנו יעבוד טוב גם עוד חודש וגם עוד שנה" [40:25] — כלומר ניטור שוטף ותהליך מובנה לתיקון, בין אם מדובר בהחלפת מודל, בשינוי prompt או בכל תיקון אחר.
- **נרמלו ואכפו את התוצר, אל תסמכו על פורמט שהמודל בחר בעצמו.** זו ההמלצה המפורשת של Aidoc: "תשקיעו כל מה שאתם צריכים כדי לאכוף, להגביל ולנרמל את תוצאות המודלים שאתם משתמשים בהם. תשלטו בתוצאה" [38:28].
- **מדדו באופן שוטף, ושתפו את התוצאות היכן שזה מתאים.** "יש לזה באמת חשיבות ממש גדולה לקרדביליות ולאמון שתקבלו" [38:28].

> **Dr. Yonatan Molad-Hayo, [38:28] —** תזכרו שהפצת מודל AI זה לא הסוף, זה באמת רק ההתחלה.

16. מילון מונחים

מונח	הסבר כפי שהוצג בסשן
Amazon Bedrock	שירות מנוהל שמנגיש מודלים מספקים שונים דרך API אחד, בתשלום לפי צריכת טוקנים, תחת מעטפת האבטחה והקומפליינס של AWS.
Bedrock Evaluations	כלי להשוואת מודלים, בשתי שיטות: Human Evaluation (דירוג ידני של מומחה תוכן) ו-Automatic Evaluation (הרצת prompt dataset מול מטריקות מובנות או custom).
Advanced Prompt Optimization	כלי שמשכתב prompt אוטומטית ומשווה עד חמישה מודלים במקביל מול ground truth, ומחזיר גם prompt מותאם וגם מדדי עלות ו-latency.
RAG	Retrieval Augmented Generation — שליפת מידע רלוונטי ממאגר והזרמתו למודל בזמן השאלה, במקום אימון או fine-tuning.
Bedrock Knowledge Bases	מימוש מנוהל של פייפליין ה-RAG: ingestion, retrieval ו-augmentation, עם API לחיפוש סמנטי ולמענה על שאלות.
Managed Knowledge Bases	הכרזה חדשה: קונקטורים ל-S3/SharePoint/Google Drive, managed store, smart parsing, אינטגרציה עם AgentCore, ו-Agentic Retrieval.
Agentic Retrieval	פיצול שאלה מורכבת לשאלות פשוטות שמורצות ברצף, עד להרכבת תשובה מלאה.
AWS Context	שירות ב-preview שבונה knowledge graph אוטומטי מעל דאטה ארגוני (structured ו-unstructured), עם מטא-דאטה ב-Apache Iceberg, שכבת governance, וחשיפה דרך Agentic Search API או MCP.
Web Grounding & Web Search	גישה מנוהלת למידע מהאינטרנט: עשרות מיליארדי מסמכים מאונדקסים בתוך AWS ומתרעננים בתדירות גבוהה, בלי egress מהרשת הפנימית.
Bedrock Guardrails	שכבת פוליסה שנבדקת פעמיים — על ה-prompt לפני המודל ועל התשובה אחריו. כוללת content filters, denied topics, PII filters, word filters, grounding checks ו-reasoning checks.
Bedrock AgentCore	אבני בניין לפריסת agents בפרודקשן: ניהול זהות והרשאות, session isolation, וזיכרון לטווח קצר וארוך.
Analyzer (Aidoc)	רכיב ייעודי לפתולוגיה אחת, שמממש עץ החלטות המשלב חוקים דטרמיניסטיים, מודלי NLP פנימיים ומודלי LLM, ומחזיר תוצר סכמטי מנורמל.
NLP Engine (Aidoc)	שכבת אורקסטרציה מבוססת Kubernetes שמריצה את ה-Analyzers בסקייל, עם caching, retries ובקרת עלויות.
NLP Live (Aidoc)	זרוע event-driven שקולטת דוחות רדיולוגיים ב-real time, מנתבת אותם ל-Analyzers המתאימים ושומרת את התוצאות ל-Data Warehouse.
Sensitivity / Specificity / PPV / NPV	מדדי ביצוע סטטיסטיים שבהם Aidoc מודדים כל מודל בכל בית חולים; PPV ו-NPV הם ערכי הניבוי החיובי והשלילי.
False positive / False negative	התראת שווא על ממצא שאינו קיים, לעומת אי-התראה על ממצא אקוטי שכן הופיע בסריקה — התרחיש החמור מבין השניים.