

סוכנים בסקייל על Bedrock AgentCore

סיכום ששן DEM304 — AWS Summit Tel Aviv 2026

מידן נאסי וסניר כהן, Solutions Architects, AWS
10 בספטמבר 2026 | משך: 25 דקות | הופק מהקלטת הששן
סיכום מאת קובי אלמוג

על המסמך הזה

המסמך מסכם את הסשן DEM304 מתוך AWS Summit Tel Aviv 2026, בהתבסס על ההקלטה המלאה שפורסמה באתר הסאמיט. הוא נבנה מתמלול אוטומטי של ההקלטה ומהסליידים שחולצו מהווידאו עצמו, כך שאפשר לחזור לתוכן בלי לצפות שוב בעשרים וחמש הדקות. חותמות הזמן בגוף המסמך מפנות לנקודה המדויקת בהקלטה.

איך לקרוא את זה

- **פרק 1 — תקציר מנהלים:** חמש דקות קריאה. מה הוצג, ומה רלוונטי לארגון פיננסי.
- **פרקים 2-4 — הרקע:** האתגרים של agents בפרודקשן, עולם ההונאות, ותבניות האורקסטרכיה.
- **פרקים 5-7 — הדמו:** הארכיטקטורה, שתי החקירות שרצו על הבמה, והקוד שמאחוריהן.
- **פרק 8 — מה לוקחים:** מסקנות ונקודות להמשך.

הערת מקור המסמך הופק מתמלול אוטומטי (Whisper large-v3-turbo) של ההקלטה. התמלול מדויק בקטעים המקצועיים, אך שמות ומונחים לועזיים עלולים להופיע בשיבוש, והוא תוקן ידנית מול הסליידים. הציטוטים אומתו מול ההקלטה בחותמת הזמן המצוינת. מספרים שמופיעים על הסליידים צוינו ככאלה.

1. תקציר מנהלים

סביר כהן ומידן נאסי, שני AWS Solutions Architects מ-AWS, לקחו תהליך עסקי אחד — חקירת הונאות בכרטיסי אשראי — והראו מה קורה כשמכניסים אליו מערכת מרובת סוכנים. הסשן לא עסק בשאלה האם agents עובדים, אלא בשאלה איך מריצים אותם בסקייל בלי לאבד שליטה: ביצועים, אבטחה, וממשל.

- **הבעיה היא לא המודל, אלא צוואר הבקבוק האנושי.** מודל ML קלאסי מסמן עסקאות חשודות בפחות מ-100 מילישניות, ומה שנתקע זה החוקר האנושי שצריך להכריע. סביר: "בסוף הבטלנק זז מצד לצד" [4:58].
- **הפתרון שהוצג הוא תוספת, לא החלפה.** המודל הקלאסי נשאר במקומו; הרכיב האג'נטי נכנס רק בנקודת ה-escalate, מוסיף הקשר ומחזיר החלטה מנומקת.
- **התבנית: Agents as tools. orchestrator.** עם מודל חזק מפרק את הקייס, וכל מומחה מקבל system prompt, כלים והקשר משלו. המומחים חשופים כ-MCP tools דרך AgentCore Gateway.
- **ההפרדה היא מה שתפס את ההונאה.** בחקירה שרצה על הבמה, שני מומחים החזירו "נקי" — והמומחה שהחזיק את הפוליסות הארגוניות הוא שזיהה סדרת עסקאות מתחת לסף הדיווח והפך את ההחלטה.
- **התשתית מנוהלת.** אותו קוד רץ לוקאלית, ב-EKS או ב-AgentCore Runtime — ההבדל הוא decorator אחד ושתי פקודות. מידן: "תנו ל-AWS להריץ את ה-agents שלכם" [18:27].

רלוונטיות לבנק

התרחיש שהוצג הוא תרחיש בנקאי במובהק: עסקאות מסומנות, מדיניות ארגונית בתוך Knowledge Base, חוקר אנושי כמשאב יקר, ודרישה לנימוק מתועד לכל החלטה. שלושת המרכיבים שהופכים את זה לשישים בארגון מפותח — הרשאות מוגבלות לכל סוכן, מסלול ביקורת מלא, ותיעוד הנימוק — הוצגו כחלק מהתשתית ולא כתוספת.

2. למה agents בפרודקשן זה קשה

סניר פתח בשלוש משפחות של אתגרים. חלקם מוכרים מהעולם הקלאסי, וחלקם חדשים לגמרי.



האתגרים כפי שהוצגו: ביצועים, סקייל, אבטחה וממשל — ומעליהם הרצת agents בפרודקשן

אי-דטרמיניזם

"נון-דטרמיניזם הוא דבר שהוא מובנה בתוך המערכות האג'נטיות וב-AI בכלל, ואנחנו מאמצים אותו. זאת אומרת זה דבר חיובי, אבל הוא מביא איתו אתגרים" [1:21]. ההשלכה המעשית: "שתי בקשות זהות אחת אחרי השנייה יכולות להתנהל בצורה אחרת לחלוטין — מבחינת נתיב הפעולה, מבחינת ה-reasoning, מבחינת הכלים שהוא בוחר להשתמש" [1:32]. בסקייל זה משפיע ישירות על תכנון הקיבולת, כי קשה לדעת כמה compute הסשן הבא יצרוך.

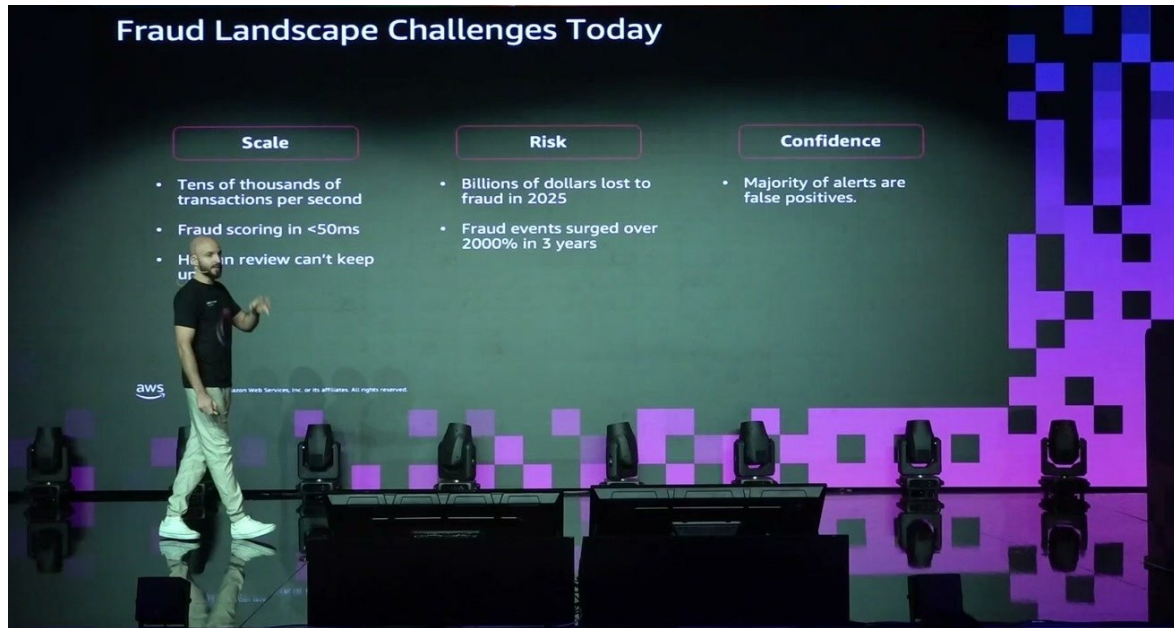
ביצועים

reasoning ארוך במיוחד, או כלי חדש שמונגש לסוכן, משנים את ה-latency של המערכת כולה [1:55]. זה לא תקלה — זו התנהגות תקינה של מערכת שמחליטה בעצמה כמה לעבוד.

אבטחה וממשל

כאן נכנסים דברים שלא היו קודם: prompt injection, ובעיקר שאלת הזהות. "אנחנו פתאום מציגים סוכן ונותנים לו to act on behalf of a user — יש לזה השלכות מאוד גדולות בהיבט ה-identity" [2:15]. ובצד הממשל: "אנחנו רוצים באמת לדעת איזה נתיב פעולה אותו סוכן בחר, למה הוא עשה את זה, מתי, איזה כלים הוא השתמש" [2:24]. נוסף על כך, הצורך בסביבות ריצה מבודדות כדי למנוע זליגת מידע בין סשנים [2:30].

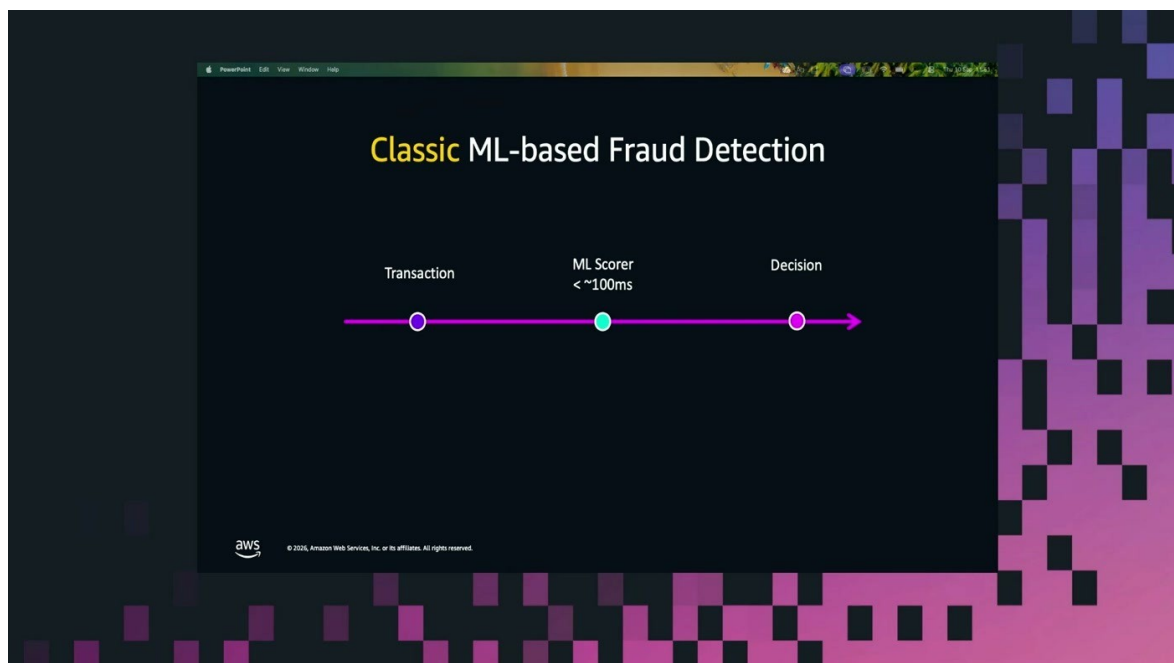
3. עולם הבעיה: חקירת הונאות



תמונת המצב שהוצגה על הבמה — סקיייל, סיכון, ורמת ודאות

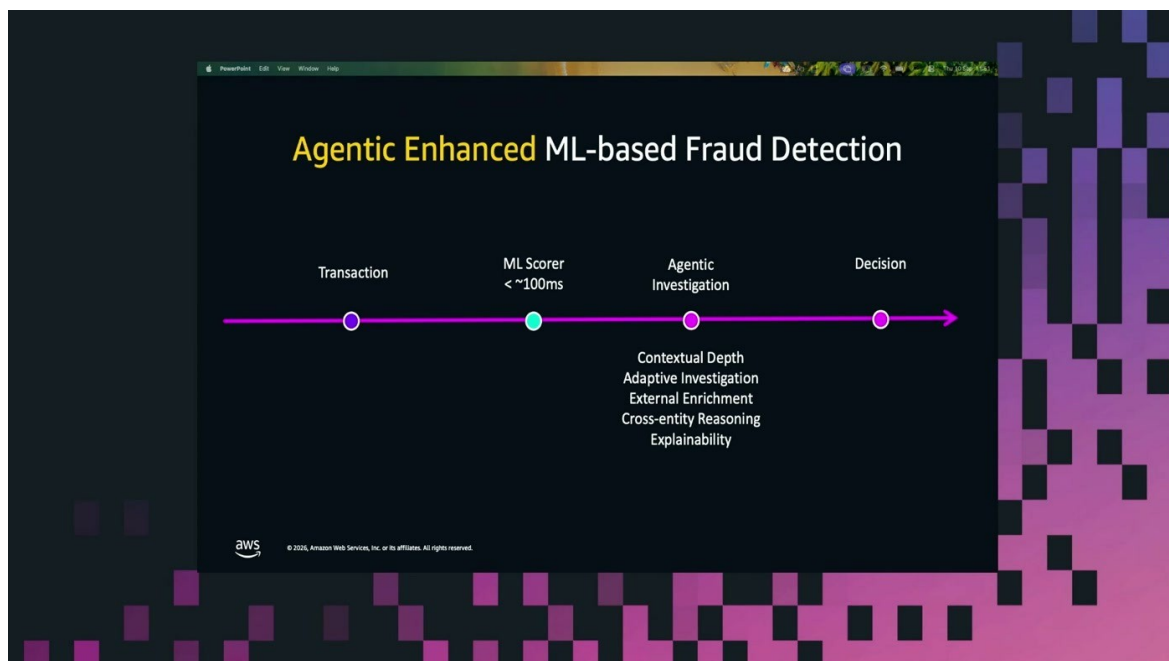
המספרים שעל הסלייד: עשרות אלפי עסקאות בשנייה, דירוג הונאה בפחות מ-50 מילישניות, ובצד הסיכון — מיליארדי דולרים שאבדו להונאה ב-2025, ועלייה של מעל 2000% באירועי הונאה בשלוש שנים. והנתון שמסביר את הכול: רוב ההתראות הן false positives.

התהליך הקיים פשוט: אובייקט עסקה נכנס למודל ML קלאסי, מקבל score, ולפי סף מוחלט — אישור, דחייה, או העברה לחוקר אנושי [4:35]. ככל שהנפח גדל, החוקר האנושי הופך למשאב הצר.



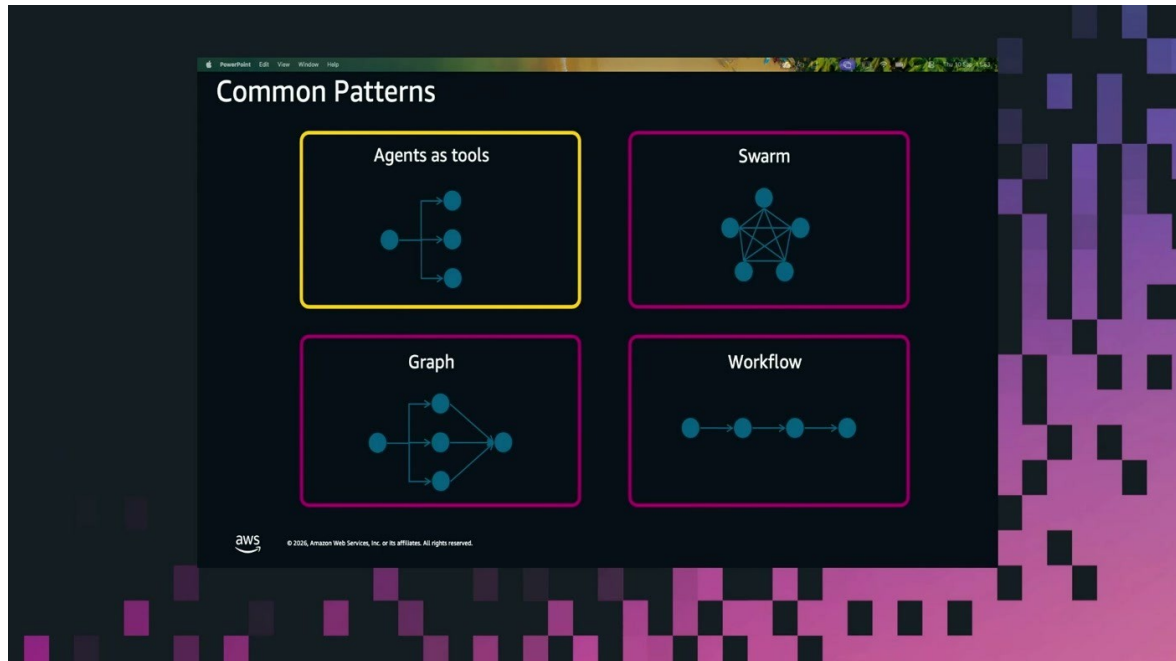
התהליך הקלאסי: עסקה, מדרג ML בפחות מ-100 מילישניות, החלטה

ההצעה לא נגעה במודל. היא הוסיפה שלב: במקום שהחוקר יקבל קייס גולמי, סוכן אג'נטי מפרק את הבקשה לתתי-משימות, מוסיף הקשר — מה קורה במערכת כרגע, מה אומרת המדיניות הארגונית — ומחזיר מסקנה מנומקת [5:31].



התוספת האג'נטית: עומק הקשרי, חקירה מסתגלת, העשרה חיצונית, הסקה חוצת-ישויות והסברתיות

4. תבניות אורקסטרציה



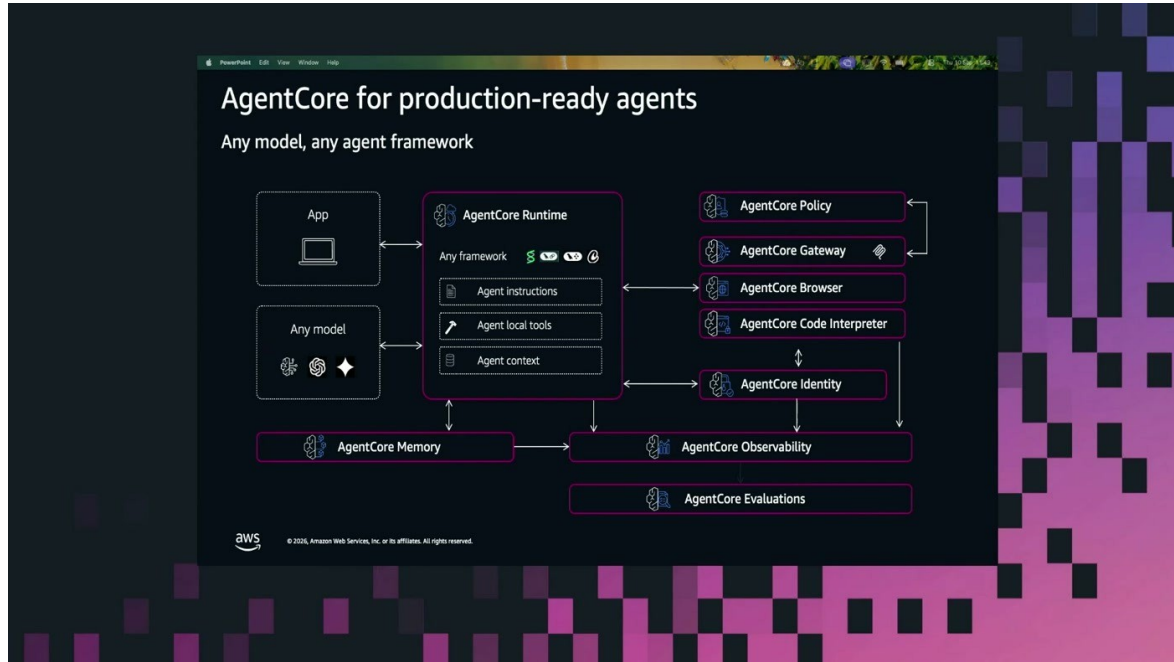
ארבע התבניות שהוצגו; המסגרת הצהובה מסמנת את זו שנבחרה לדמו

סניר הציג ארבע תבניות והמליץ לקרוא עליהן לפני שבוחרים: "כל אחת עונה על בעיה אחרת" [6:35]. לדמו נבחרה Agents as tools.

תבנית	מי קובע את מסלול הביצוע	מתי מתאים
Agents as tools	אורקסטרטור מרכזי שקורא למומחים כמו לפונקציות	חלוקת אחריות ברורה, ביקורת פשוטה
Swarm	הסוכנים עצמם, עם העברת שליטה והקשר משותף	משימה פתוחה שדורשת שיתוף פעולה
Graph	גרף דטרמיניסטי שהמפתח הגדיר מראש	תהליך מובנה עם תנאים
Workflow	תלויות בין משימות, ריצה במקביל	תהליך חוזר וארוך עם שלבים

הנימוק לבחירה: "אורקסטרטור יכול להיות עם מודל חזק יותר", שמפרק לתתי-משימות ומנגיש לכל sub-agent הקשר ספציפי, system prompt ספציפי וכלים ספציפיים [6:57]. ומידן הוסיף בהמשך: "האורקסטרטור אג'נט הוא הראוטר החכם שלי" [13:28], ו"הטולים שלו בפאטרן הזה ספציפית — הם גם אג'נטים" [13:44].

5. AgentCore — הרכיבים שבשימוש



Amazon Bedrock AgentCore: כל רכיב עונה על אחד מהאתגרים מפרק 2

רכיב	מה הוא נותן	איך שימש בדמו
Runtime	סביבה מנוהלת, serverless וסקיילבילית להרצת הלוגיקה; אגנוסטית לפריימוורק ולפרוטוקול	הריץ את האורקסטרטור ואת כל המומחים
Gateway	נקודת גישה אחודה לכלים, שירותי צד שלישי וסוכנים אחרים, בפרוטוקול MCP	חשף כל sub-agent כ-MCP tool לאורקסטרטור
Code Interpreter	הרצת קוד שהסוכן כתב, בסביבה נקייה בכל ריצה	המומחה הסטטיסטי הריץ חישוב סטיית תקן
Browser	דפדפן בסביבה מבודדת	בדיקת אתר הסוחר, כולל תקינות ה-TLS
Observability	טלמטריה אוטומטית אל CloudWatch	מדדי סשנים, קריאות ושגיאות

לתשומת לב Policy ו-Identity הוזכרו כרכיבים משמעותיים אך לא הודגמו מפאת זמן, והופנו לסשנים אחרים בסאמיט [4:01]. בארגון פיננסי דווקא הם הרכיבים שמצדיקים בדיקה נפרדת.

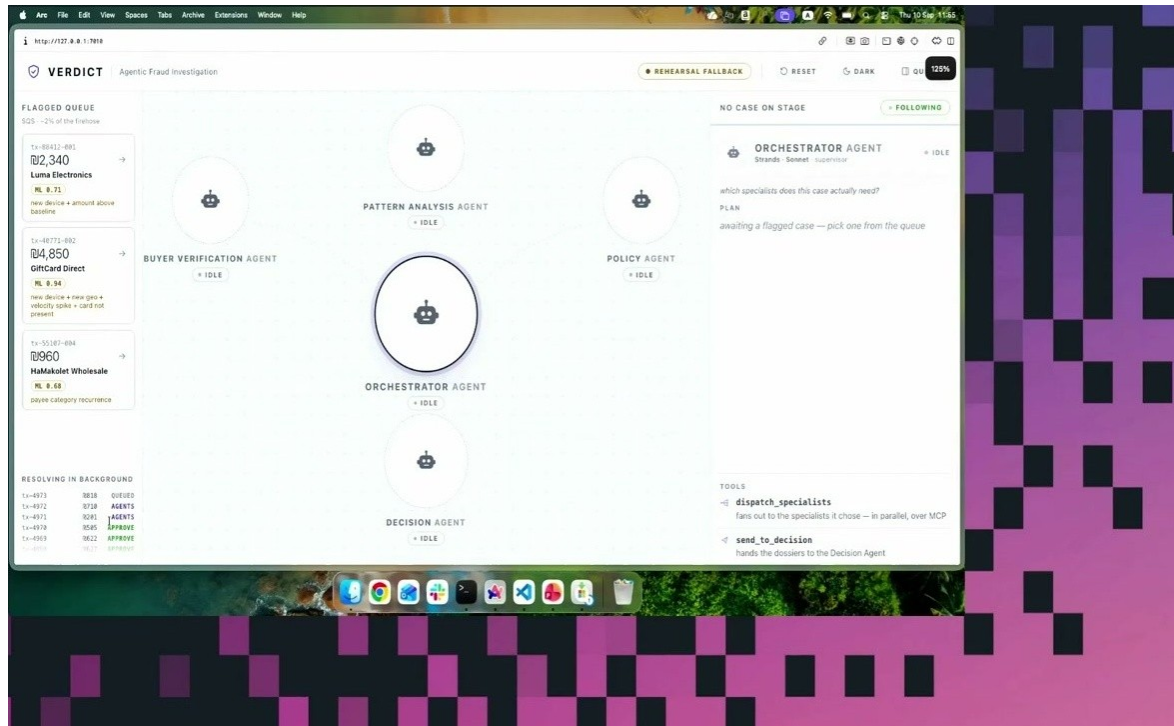
6. הארכיטקטורה של הדמו

הזרימה כפי שהוצגה [7:19]: עסקאות מסומנות נכנסות לתור SQS. פונקציית Lambda שואבת מהתור ומפעילה את האורקסטרטור, שרץ על AgentCore Runtime. האורקסטרטור מחליט אילו מומחים הקייס צריך, ופונה אליהם דרך AgentCore Gateway — "שימו לב, זה אג'נטים בפני עצמם, אבל הם מונגשים כ-tools לאורקסטרטור שלנו, דרך אותו MCP פרטוקול" [8:15].

המומחים

- **Policy Agent** — גישה ל-Knowledge Base עם המדיניות הארגונית; מחזיר התאמה לחוקה ספציפית.
 - **Pattern Analysis Agent** — שולף 90 עסקאות אחרונות, כותב קוד ומריץ אותו ב-Code Interpreter.
 - **Buyer Verification Agent** — פונה לשירות due diligence פנימי, ובודק את אתר הסוחר בדפדפן.
 - **Decision Agent** — מקבל את סיכום האורקסטרטור ומכריע.
- הפריימוורק בדמו הוא Strands, "אבל זה יכול להיות LangGraph או [7:42] ADK". בפועל שני מומחים נכתבו בשני פריימוורקים שונים בכוונה, כדי להראות שה-Runtime אדיש לבחירה.

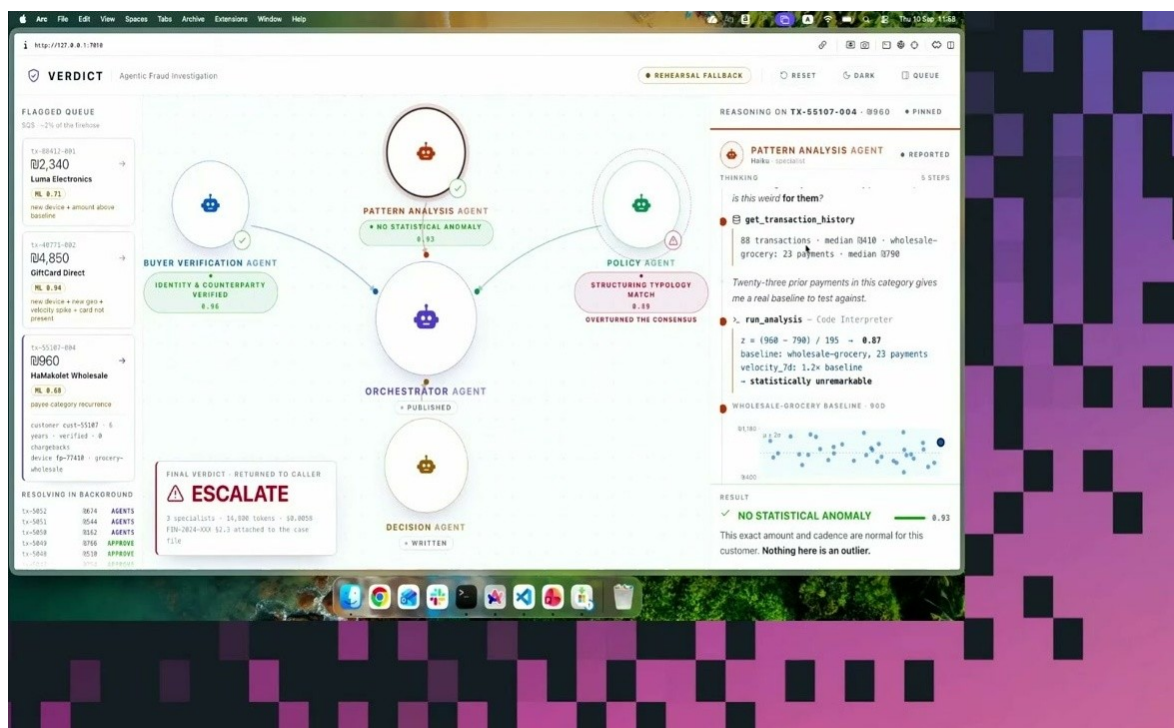
7. הדמו: שתי חקירות



הממשק לפני הרצה: תור העסקאות המסומנות משמאל, הסוכנים במרכז, והנימוק מימין

חקירה ראשונה — העברה לחוקר אנושי

האורקסטרטור קיבל עסקה בסך \$960 אצל סוחר סיטונאי, מלקוח ותיק. הוא עשה planning, פירק לתתי-בעיות, והפעיל את שלושת המומחים במקביל. Pattern Analysis שלף 90 עסקאות, ייצר קוד שמחשב סטיית תקן, והחזיר שאין אנומליה. Buyer Verification חזר נקי. ה-Policy Agent פנה ל-Knowledge Base, השווה מול הפוליסות, וזיהה הפרה: סדרת עסקאות שכל אחת מהן מתחת לסף הדיווח, בחלון זמן דחוס. ה-Decision Agent ראה שאחד המומחים החזיר ממצא, והכריע: העברה לחוקר אנושי.



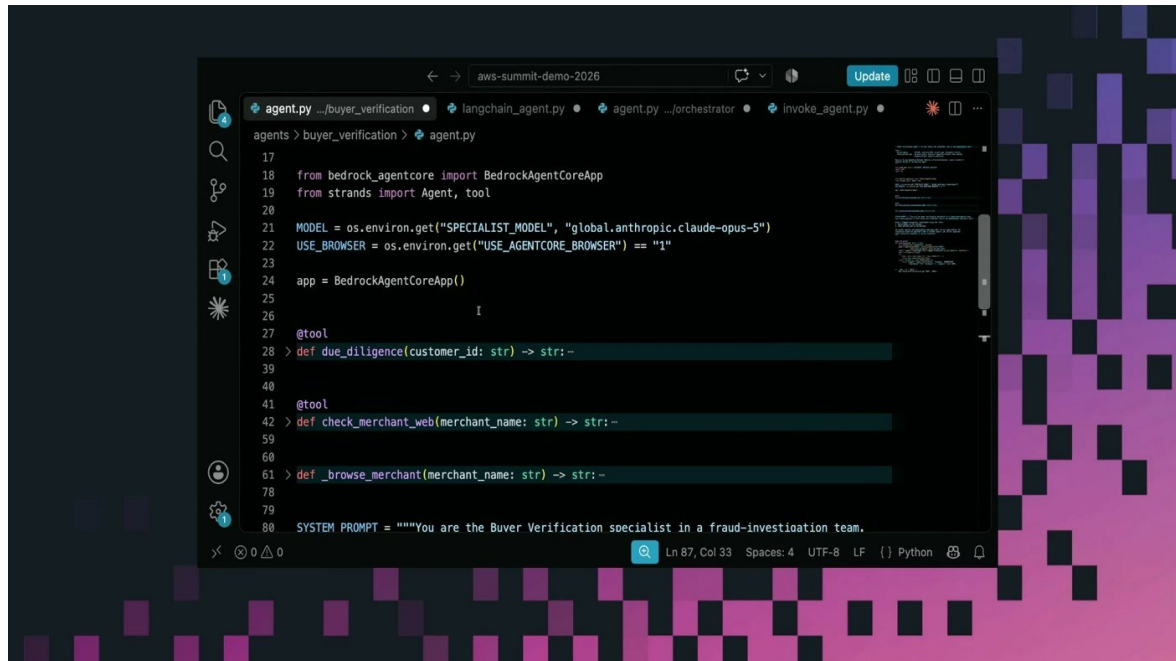
המסך בסיום החקירה: מימין הנימוק של מומחה הדפוסים, ומשמאל ההחלטה — ESCALATE

המסקנה זה הרגע המעניין בסשן. שני מומחים אמרו "אין ממצא", והמומחה עם הידע הרגולטורי הפך את התוצאה. ההפרדה בין המומחים היא לא אלגנטיות ארכיטקטונית — היא מה שאיפשר לממצא הזה לצוף.

חקירה שנייה — דחייה

בקייס השני שלושת המומחים החזירו ממצא בעייתי, והבולט שבהם היה של Buyer Verification: "הבייר ביצע רכישה לפני 41 דקות בתל אביב, והרכישה הנוכחית מבוקרשט. איך הוא הגיע מתל אביב לבוקרשט ב-41 דקות?" [14:34]. ה-Decision Agent החזיר דחייה. מידן סיכם את העיקרון: "אנחנו עושים Context Enrichment — הוספנו הקשר וזיהינו בעיות יותר רחבות" [14:46].

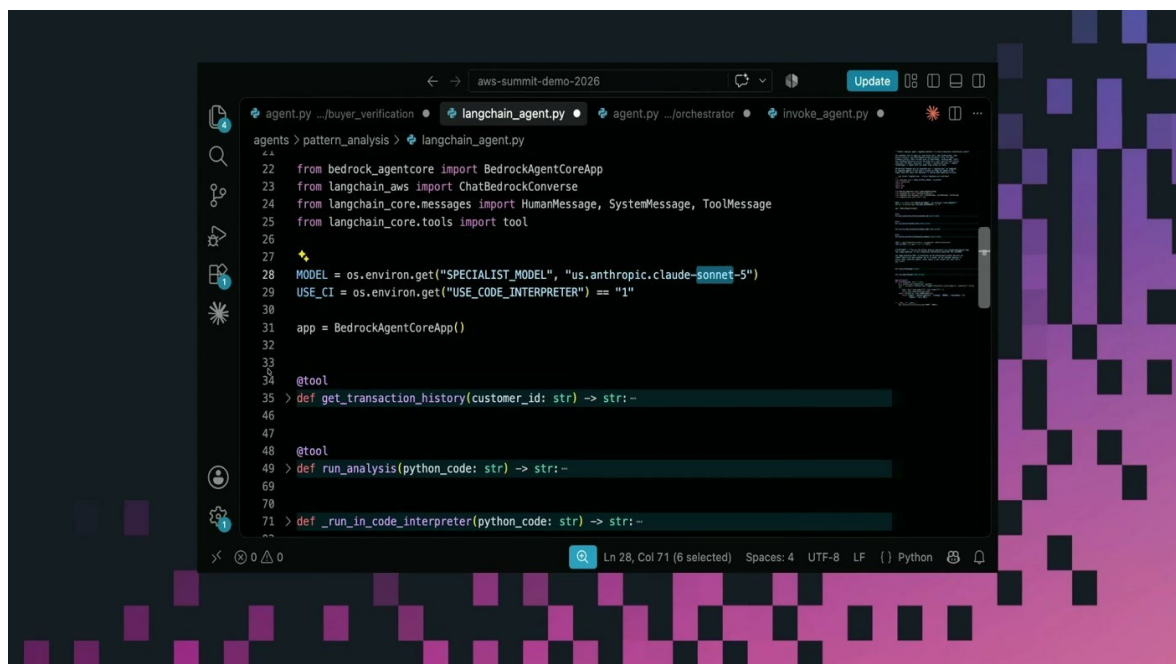
8. מאחורי הקלעים: הקוד והתשתית



```
17 from bedrock_agentcore import BedrockAgentCoreApp
18 from strands import Agent, tool
19
20
21 MODEL = os.environ.get("SPECIALIST_MODEL", "global.anthropic.claude-opus-5")
22 USE_BROWSER = os.environ.get("USE_AGENTCORE_BROWSER") == "1"
23
24 app = BedrockAgentCoreApp()
25
26
27 @tool
28 > def due_diligence(customer_id: str) -> str:--
29
30
40
41 @tool
42 > def check_merchant_web(merchant_name: str) -> str:--
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61 > def browse_merchant(merchant_name: str) -> str:--
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80 SYSTEM PROMPT = """"You are the Buver Verification specialist in a fraud-investigation team.
```

המומחה לאימות קונים: Strands, כלי due diligence ובדיקת אתר הסוחר בדפדפן

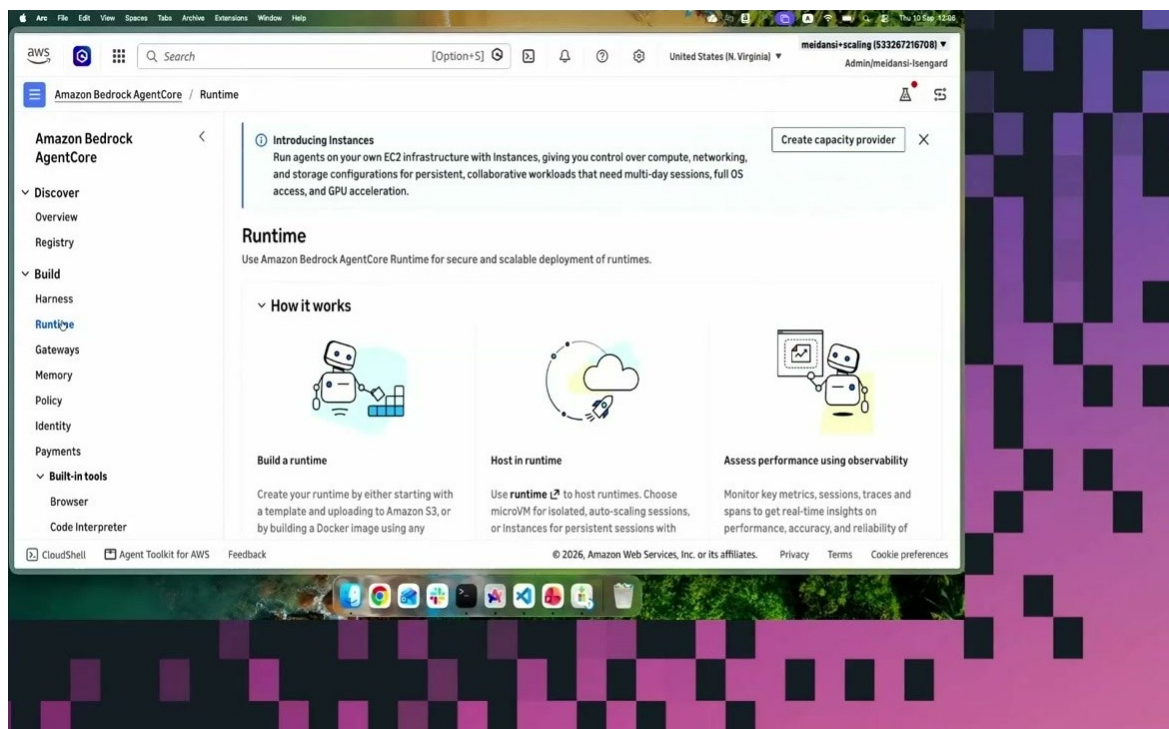
שורת הייבוא הראשונה היא כל הסיפור: הקוד הוא קוד Python רגיל עם כלים מסומנים ב-decorator, והמעבר ל-AgentCore הוא עטיפה של ה-entry point ואחריה שתי פקודות: configure ו-[17:16] launch. "אותו קוד יכול לרוץ היום ב-EKS בקונטיינר או לוקאלית — ההבדל היחיד הוא הדקורטור" [17:46].



```
22 from bedrock_agentcore import BedrockAgentCoreApp
23 from langchain_aws import ChatBedrockConverse
24 from langchain_core.messages import HumanMessage, SystemMessage, ToolMessage
25 from langchain_core.tools import tool
26
27
28 MODEL = os.environ.get("SPECIALIST_MODEL", "us.anthropic.claude-sonnet-5")
29 USE_CI = os.environ.get("USE_CODE_INTERPRETER") == "1"
30
31 app = BedrockAgentCoreApp()
32
33
34 @tool
35 > def get_transaction_history(customer_id: str) -> str:--
36
37
38
39
40
41
42
43
44
45
46
47
48 @tool
49 > def run_analysis(python_code: str) -> str:--
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71 > def _run_in_code_interpreter(python_code: str) -> str:--
72
```

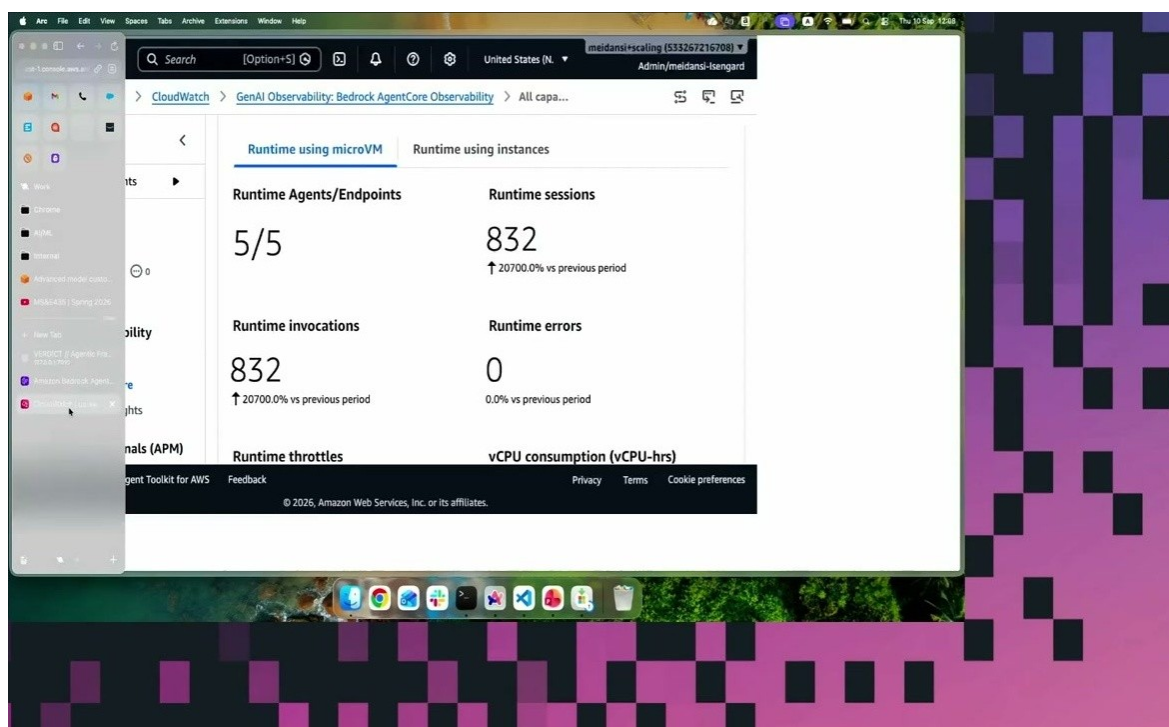
מומחה ניתוח הדפוסים נכתב ב-LangChain — במכוון, כדי להראות שה-Runtime אגנוסטי

שני המומחים רצים על מודלים שונים: המומחה לאימות קונים על Claude Opus, ומומחה הדפוסים על Claude Sonnet עם Code Interpreter. זו החלטת עלות מודעת — מודל חזק לשיפוט, מודל מהיר לניתוח מספרים.



הרצת ה-agents ב-AgentCore Runtime מתוך הקונסולה

ב-Gateway נוצרו targets, כל אחד מצביע על ה-invoke URL של מומחה ב-Runtime. מבחינת האורקסטרטור זה MCP רגיל: "התחברתי עם MCP client כמו שהייתי מתחבר לכל MCP מרוחק" [19:37]. מה שזה נותן בפועל: אפשר להחצין גם APIs קיימים בארגון כ-tools דרך אותה דלת [19:29], ולהוסיף שכבת אימות לפני הכניסה לסוכן ובקרה על מה שהוא רשאי לגשת אליו [21:00].



הטלמטריה נשפכת אוטומטית ל-CloudWatch: סטנים, קריאות ושגיאות

"אני לא צריך לדאוג לסקייל. AgentCore Runtime דואגת לסקייל הזה" [22:52].

9. מה לוקחים מכאן

- **multi-agent** זה חלוקת אחריות, לא עוד סוכן. אורקסטרטור, מומחים, policy והכרעה — ישויות נפרדות, כל אחת עם הקשר, כלים והרשאות משלה.
- **המומחה הזול עם הידע הנכון מנצח.** לא המודל הגדול ביותר תפס את ההונאה, אלא זה שהחזיק את המדיניות הארגונית.
- **Gateway** הוא דלת אחת עם שומר. כל קריאה לכלי עוברת דרך הרשאות ומסלול ביקורת, ולא דרך אמון בין רכיבים.
- **אפשר לדלג על מומחים.** אם ה-score הקלאסי לא הצביע על הקונה, האורקסטרטור לא מפעיל את מומחה האימות. מידן: "חסכנו זמן, חסכנו טוקנים, חסכנו [16:04] latency".
- **התבנית ניתנת להחלפה.** "אם אני משנה משהו בפוליסי אג'נט, זה לא בהכרח מהותית ישפיע על כל הארכיטקטורה" [15:31].

נקודות להמשך

נושא	למה זה חשוב	חזרת זמן
AgentCore Policy ו-Identity	לא הודגמו בסשן; הרכיבים שמכריעים היתכנות בארגון מפורק	21:00, 4:01
בחירת מודל לכל מומחה	השפעה ישירה על עלות ועל latency	16:59
Observability ו- evaluations	איך יודעים שהסוכן בחר נכון, בלי שגיאה בלוג	22:17
הפעלת מומחים סלקטיבית	החיסכון האמיתי נמצא שם	15:43

10. מונחון

מונח	משמעות
Agents as tools	תבנית שבה סוכנים מומחים מונגשים לאורקסטרטור כאילו היו כלים
MCP	פרוטוקול אחיד לחיבור מודל לכלים ולשירותים חיצוניים
AgentCore Runtime	סביבת הרצה מנוהלת לסוכנים, עם בידוד בין ששנים
AgentCore Gateway	נקודת גישה אחודה שמחצינה כלים וסוכנים בפרוטוקול MCP
Strands	SDK בקוד פתוח לבניית סוכנים, שנכתב במקור ב-AWS
Knowledge Base	מאגר מסמכים לשליפה סמנטית, כאן מדיניות ארגונית
Structuring	פיצול סכום לעסקאות קטנות כדי לעקוף סף דיווח

