

23/6/20

עיצוב ופיתוח מערכות נתוני עתק ומחשוב ענן  
פרויקט סיום תש"פ

כללי:

בתקופה הנוכחית, בנוסף לנתונים המופקים ממערכות תפעוליות, מאותרים ונאספים נתונים רבים ממגוון מקורות כבסיס לניתוח, גיבוש תובנות, אימוץ אסטרטגיות וקבלת החלטות מבוססת ראיות.

מערכות Big Data Analytics דורשות משאבים ארגוניים רבים ולכן שימוש בספקי שירותי ענן לאחסון ועיבוד הם פרקטיקה נפוצה. בתכנון מערכת מבוצרת מסוג זה אין מניעה לטוות מגוון שירותים ממספר ספקים, בשילוב שירותים שמקורם בארגון עצמו, לכלל פתרון שלם.

בפרויקט הנוכחי ניצור פתרון לעיבוד ותצוגת נתונים בהשראת [תבנית Lambda](#), אם כי לצורך מיקוד המאמצים בסמסטר זה הדימיון [לתבנית Kappa](#) קרוב יותר לדרישות החובה.

מהות הפתרון:

מערכת הבנויה משלוש תת מערכות שיחידיו מאפשרות ניטור שיחות נכנסות ל- Call Center בגישת Near Real Time ובאמצעות Dashboard המציג מדדי מפתח, וכן מאפשרת ניתוח היקפי שיחות עבר ומאפייניהם.

דרישות פונקציונליות:

1. המערכת תקבל נתוני שיחות Call Center ותדאג לאחסנם, לעבדם ולהציגם (תמונה 1 כדוגמא).
2. הנתונים המיוצגים במערכת :
  - מספר שיחות הממתינות למענה.
  - המצב כהקשר (חירום, שגרת חירום, רגיעה).
  - זמן התחלת שיחה.
  - משך שיחה.
  - עיר המתקשר.
  - נושא (תרופות, מזון, מים, מיגון, מידע, פינוי, טיפול רפואי).
  - שפת המתקשר (עברית, אנגלית, אמהרית, רוסית, תאילנדית).
  - מין המתקשר.
  - גיל המתקשר.
3. המערכת תציג ב- Real-Time Dashboard את מספר השיחות הממתינות כרגע וזמן המתנה ממוצע בעשר הדקות האחרונות באמצעות גראפים \ Widgets.
4. המערכת תאפשר להציג בטבלה ובגראף את מדדי מספר השיחות הממתינות וזמני המתנה מתחילת היום ברמת אגרגציה של 5 דק'.
5. המערכת תאפשר צפייה במדדים סיכומיים לגבי היום (טבלה) : מספר שיחות מכל יישוב, מספר שיחות בכל נושא.
6. המערכת תאפשר צפייה במדדים סיכומיים לגבי היום (גראפים): התפלגות השיחות על פי סוג הבקשה ולפי שפה.
7. כל נתוני השיחות יישמרו לצורך תחקורים וניתוח.

8. אופציונאלי: המערכת תאפשר למידה מהנתונים על בסיס שאלת חקר שתגדירו ( איתור אסוציאציה בין סוגי מימדים כדוגמת קבוצת גיל או מגדר לסוג הבקשה, אשכולי שיחות או חיזוי לגבי סוגי בקשות בבוקר, בצהרים ובערב (בהנתן הקשר מצב נתון)).
9. אופציונאלי: לאחר למידת המודל ניתן יהיה לפנות אליו כשירות או לייצאו כקוד למערכת ולתשאלו לגבי נתונים חדשים (לדוגמא מה נפח השאלות הצפוי בשפה מסוימת בבוקר הנוכחי אם אנו בשגרת חירום בצפון ).

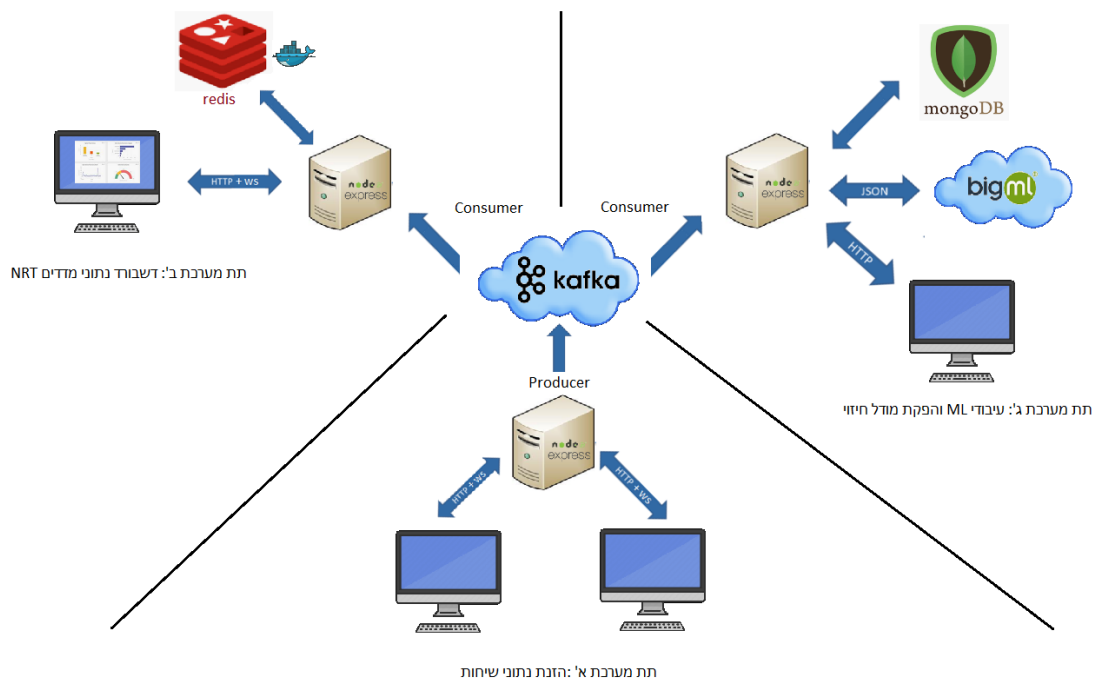
#### דרישות לא פונקציונאליות:

1. המערכת תפותח בתפיסה מונחית שירותים בגישת [Micro Services](#).
2. חלקו את המערכת לשלוש תת מערכות והשתמשו ב- Kafka כ- Message Broker לאינטגרציה ביניהם.
3. שלוש תת מערכות הן:
  - א': תת מערכת קליטת נתונים.
  - ב': תת מערכת Dashboards ונתוני זמן אמת.
  - ג': תת מערכת אחסון כל נתוני השיחות ואופציונאלית יצירת מודל חיזוי באמצעות למידת מכונה.
4. השתמשו בתשתיות Node.js/express, Kafka, Redis, MongoDB וחבילות D3.js או Chart.js לפיתוח הפתרון.
5. Kafka מומלץ להריץ בענן ולהשתמש בספק: <https://www.cloudkarafka.com/>
6. Redis מומלץ להריץ במסגרת Docker עם ה-image [https://hub.docker.com/\\_/redis](https://hub.docker.com/_/redis)
7. MongoDB ניתן להריץ לוקאלית, ע"ג Docker Image או משירות ענן כגון: <https://www.mongodb.com/>
8. תת מערכת ב' זוכרת רק נתוני תאריך נוכחי (מחצות עד חצות) על ידי מסד נתונים מסוג Redis ומתאפסת בחצות או על ידי בקשה מהמשתמש.
9. תת מערכת ג' זוכרת את כל נתוני השיחות היסטורית ע"ג MongoDB.
10. מעבר לטעינה ראשונית באמצעות בקשת HTTP, תת מערכת א' ותת מערכת ב' מעדכנים נתונים בינם לבין הדפדפן באמצעות פרוטוקול WebSockets בעזרת חבילת Socket.io.
11. ארגון השירותים בסביבת Node.js תתבסס על תבנית MVC ( השתמשו ב-express כמרכיב View המחולל תצוגה ).
12. אופציונאלית (לא חובה): תת מערכת ג' תממש למידה מנתונים על ידי שימוש בשירותי BigML.com וחבילת Node המתאימה.

Total Waiting calls:

Answer Call						
Start Time	City	Topic	Language	Gender	Age	End
25-Jun-2020, 9:03	ירושלים	תרופות	עברית	גבר	56	סיום
25-Jun-2020, 9:03	ירושלים	מזון	אנגלית	אישה	14	סיום
25-Jun-2020, 9:03	ירושלים	מים	אמהרית	אישה	26	סיום
25-Jun-2020, 9:04	ירושלים	מיגון	רוסית	גבר	67	סיום
25-Jun-2020, 9:04	ירושלים	מידע	תאילנדית	גבר	32	סיום
25-Jun-2020, 9:04	ירושלים	פינוי	עברית	גבר	33	סיום
25-Jun-2020, 9:04	ירושלים	טיפול רפואי	עברית	אישה	45	סיום

תמונה 1: מסך קליטת נתונים לדוגמא



דיאגרמה 1 : שלושת תת המערכות בפתרון

#### הערות:

1. ההגשה בזוגות אך ניתן גם בשלישייה.
2. נא להרשם להגשה כבר עכשיו [בלינק הזה](#).
3. מימוש למידת מכונה בתת מערכת ג' הינה אופציה למגישים בזוגות אך חובה לשלישייה.
4. מוזמנים להשתמש בדוגמאות הקוד במודל אך יש להתאימם לפתרון שלכם.
5. את הקוד ותאור הפרויקט יש לשמור על גבי GitHub.
6. תאריך ההגשה : 30.8 – כנראה באמצעות זום.
7. בנוס: פריסת תת מערכת א' ע"ג ספק שירותי אירוח כגון <http://heroku.com> כמתואר כאן: <https://www.geeksforgeeks.org/deploying-node-applications/?ref=rp>

בהצלחה.