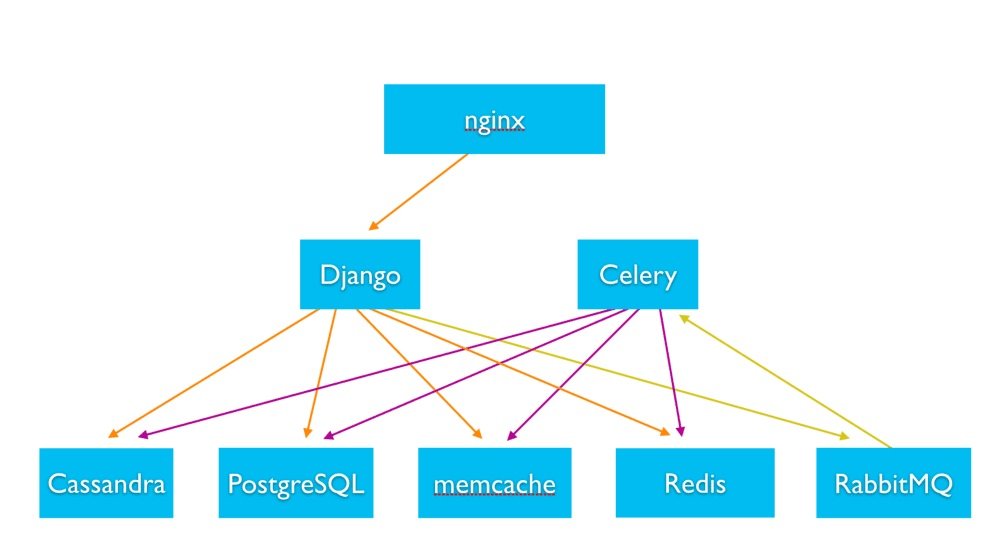
**שפות תכנות והידור תרגיל 1 – אלעד ישראל 313448888 ונרקיס שלו קרמיזי 205832447**

1. ארכיטקטורת Instagram:

הבעיה העיקרית שעימה צריך להתמודד באפליקציה בעלת יותר מביליון משתמשים, היא איך להתמודד עם כמות גדולה כזו של פניות לשרתים שלה. כאן נכנסת לתמונה nginx - שרתים המתוכנתים לטפל במספר רב של בקשות תוך צריכת משאבים מועטים, ושמירה על ביצועים טובים.

**קוד צד-השרת** של אינסטגרם נשען על Django- תשתית תוכנה[[1]](#footnote-1)(framework) לכתיבת אפליקציות רשת הכתובה בPython.

בצד האחורי של התוכנה, נעשה שימוש בטכנולוגיות שונות להצגת תוכן המותאם אישית למשתמש: Cassandra (בסיס נתונים מבוזר מסוג NO-SQL לטיפול בBig Data וכתוב בשפת Java), ו-PostgreSQL, Memcache, Redis (עליהם נפרט בהמשך).

משימות א-סינכרוניות (שלא צריך לחכות שהן יסתיימו לפני מעבר למשימה הבאה) כגון שליחת התראות למשתמשים ולשירותים שונים ברקע, מתבצעות ע"י RabbitMQ ו- Celery. Celery הוא תור של משימות אסינכרוניות המבוסס על העברת מסרים בין תהליכים/אובייקטים, כאשר המתווך - מי שהתפקיד שלו הוא להעביר הודעות מהצד השולח לצד המקבל, הוא RabbitMQ.  
כאשר אפליקציה רוצה לקבל/לשלוח הודעה, היא פותחת connection מול RabbitMQ ובעזרתו היא פותחת channel (ערוץ שיחה) ומאזינה בפועל לתור (או משתמשת ב"גשר" - Exchange- מה שמאפשר מניפולציות על ההודעה, כגון שליחה למספר אפליקציות).  
בנוסף נעשה שימוש בGearman - framework המאפשרת לחלק משימות לnodeים אחרים במערכת, לדוגמא מכונות או תהליכים שיותר מתאימים לבצע משימות אלו.

**האחסון** של אינסטגרם מתבסס על 12 "shards" (שברים) - מחיצה של מידע של בסיס הנתונים, כאשר כל מחיצה מנוהלת כיחידה עצמאית ונפרדת (כבסיס נתונים נפרד). דבר זה נעשה ע"ב גישת NO-SQL לאחר שמפתחי התוכנה ראו שבסיס הנתונים שלהם נהיה עצום וזקוק לscaling בדחיפות. כך גודל הטבלאות יורד משמעותית, אך מידע מסוים יכול להימצא על כל השברים או אפילו על שבר אחד בלבד. בסיס הנתונים מנוהל ע"י PostgreSQL - מערכת לניהול relative DB.  
בנוסף נעשה שימוש ב[[2]](#footnote-2)Apache Hive לארכוב (אחסון לזמן ארוך) של הנתונים במרווחי זמן מסויימים;  
בVmtouch לניהול המידע בזיכרון במעבר בין מכונות; בPgbouncer לשמירה על חיבורים לשרת האחורי בpool כדי לעשות בהם שימוש חוזר ובכך להקפיץ את הביצועים; בRedis – בסיס נתונים המנוהל בזיכרון (ולא בדיסק) ולכן מהיר ביותר, לשמירת המידע הreal time של האפליקציה - כגון הפיד, הסשנים וכו'; וכמובן בMemcache לייעול ע"י caching של המידע במערכת.  
שכבת הcache ממוקמת פיזית בצמוד לשרתי הWeb ע"מ להימנע מlatency. מידע גלובלי משוכפל לכמה מרכזי מידע בעולם, לעומת מידע מקומי המוגבל למרכז המידע הספציפי של האזור - מה שעוזר לScalability.  
הצד האחורי של האפליקציה היה מאוחסן בהתחלה אצל AWS של אמאזון, אך הועבר למרכזי המידע של פייסבוק, מה שהקל על האינטגרציה איתם (אינסטגרם נמכרה לפייסבוק), קיצר latency, ואיפשר להשתמש בכלים מובנים של פייסבוק להתמודדות עם פריסה בקנה מידה גדול.

פיקוח על מצב האפליקציה וזמינות השירות מתאפשר הודות לMunin - כלי עזר לניטור תשתיות ורשתות ודיווח על חריגות, Pingdom לניטור חיצוני של זמינות וביצועים, וPagerDuty להתראות וטיפול בחריגות.

**חיפוש התמונות** של אינסטגרם נעשה בעזרת Unicorn- מערכת אינדקסים ומנוע חיפוש ברשת חברתית שנבנה ע"י פייסבוק, ומאפשר לדוגמא לחפש חברים של חברים או לשמור מיקומים ותיוגים. הוא עושה זאת ע"י מיפוי של תנאים (כגון "חבר של") לקבוצות של מסמכים לפי הID של המשתמשים.  
המידע השמור באינדקסים האלו מוכוון לחיפוש, בניגוד לסתם מידע השמור ע"י PostgreSQL DB.

2. שפות התוכנה שפותחו ע"י פייסבוק הן Hack ו- Thrift.

**Hack** היא שפת תכנות בעלת קוד פתוח שפותחה עבור המכונה הווירטואלית HipHop. השפה פותחה כהרחבה לPHP שהיא שפת תסריט המתמקדת בפיתוח באינטרנט וניתנת להטמעה בHTML. Hack משלימה את מחזור הפיתוח המהיר של PHP עם המשמעת שמספקת טיפוסיות סטטית, תוך הוספת תכונות רבות הנפוצות בשפות תכנות מודרניות אחרות.

בפיתוח של Hack פייסבוק רצו ליהנות מהיתרונות של טיפוסיות סטטית וגם מהיתרונות של טיפוסיות דינאמית. מערכת הטיפוסים שהם יצרו לשפה היא מסוג Gradual typing - מערכת טיפוסים שבה חלק מהמשתנים בעלי טיפוסיות סטטית וחלק מהם בעלי טיפוסיות דינאמית.

טיפוסיות סטטית היא טיפוסיות בה לכל משתנה ולכל ערך יש טיפוס. כל משתנה מטיפוס מסוים יכול להכיל רק ערכים מאותו טיפוס (או מתת-טיפוס שלו) וכל הטיפוסים נבדקים בזמן הידור .כך המהדר יכול להתריע במקרה של שגיאות. בשפות המשתמשות בטיפוסיות סטטית, עצם העובדה שתוכנית עברה הידור מעיד שלא קיימות בה שגיאות מסוגים מסוימים.

לעומת זאת, טיפוסיות דינאמית היא טיפוסיות בה הטיפוס מוגדר בזמן ריצה, ורק אז מתבצעות הבדיקות המתאימות ע"מ לוודא שלא התרחשו שגיאות. תכונה זאת מקלה על הכנסת שגיאות, וגורמת לגילוי שלהן להתרחש מאוחר למדי, אם בכלל. מצד שני, קל יותר לכתוב תוכניות פשוטות בשפות שמשתמשות בטיפוסיות דינאמית, וזמן הכתיבה קצר בהן בהרבה.

**Thrift** היא שפת הגדרת ממשק ופרוטוקול תקשורת בינארית, בעלת קוד פתוח, המשמשת להגדרה ויצירה של שירותים עבור מספר שפות. השפה יוצרת framework של הפעלת פרוצדורות מרוחקות ופותחה בפייסבוק לצורך פיתוח שירותים סקלאבילים חוצי שפה.

השפה משלבת software stack(מקבץ תוכנות) עם מנוע יצירת קוד לבניית שירותים חוצי פלטפורמה שיכולים לחבר בין יישומים הכתובים במגוון שפות ו-frameworks.

3. א. הבעיה העיקרית בכתיבת אפליקציות Web היא בדר"כ דווקא לא תכנות חלקי האתר השונים - אלא חיבורם.  
אתר מורכב מחלקים שונים הבנויים בטכנולוגיות שונות, ולגרום לחלקים השונים לעבוד יחד זה חלק גדול מהעבודה, במיוחד למתחילים שנדרשים להכיר טכנולוגיות רבות רק כדי לגרום לאתר שלהם לעבוד.   
פה Meteor נכנס לתמונה - ביצירת פרויקט חדש ב-Meteor הוא נוצר עם כל הרכיבים הדרושים מראש, ומגדיר את כולם לעבוד יחד "מאחורי הקלעים". זה מאפשר למתכנת להתעסק בתכנות ולא בהגדרות של החיבורים בין הרכיבים.

Meteor נכתב בעזרת Node.js מה שמאפשר לכתוב גם את צד השרת בJavaScript. כלומר מפתח אחד שיודע JavaScript יכול לכתוב את כל אפליקציית הWeb, ולא נדרש מפתח אחד בצד לקוח שידע JavaScript, ומפתח נוסף בצד השרת שידע Python או Ruby וכדומה... כל הדברים לעיל מאפשרים כמובן פיתוח מהיר יותר ולכן גם זול יותר.  
נוסף על כל זאת, Meteor היא Cross-platform, קלה מאוד להבנה, ובאה עם אתר בסיסי מוכן שרץ ישר לאחר ההתקנה! זה דבר מעולה למתחילים, שיכולים ישר להתחיל להתנסות בבניית אתר ללא צורך בכל הדברים "מסביב". Meteor גם דואג לעדכן את האתר בשנייה שהמשתמש מבצע שינויים ושומר אותם (שוב, חוסך התעסקות רבה בכל שינוי שמבוצע באתר).

ב. Web framework – תשתית תוכנה (פירוט בהערות שוליים בעמוד 1) המיועדת לתמוך בפיתוח אפליקציות Web, כולל web services, web API ועוד.

1. Framework הוא בסיס לפיתוח תוכנה או אתר אינטרנט שנותן את הכלים הבסיסיים למתכנת מבלי שיצטרך לעבוד יותר מדי קשה, לרוב Framework יביא דברים בסיסיים שכל אתר צריך, כמו למשל מערכת grids(טבלאות למיקום האלמנטים) ואפשרות לאתר responsive(שמתאים את עצמו לגודל המסך), שאין סיבה שיכתבו כל פעם מחדש, דוגמאות טובות ל-Frameworks הםBootstrap ו- jQuery. [↑](#footnote-ref-1)
2. תוכנה לאחסון נתונים המאפשרת שאילתות, ניתוחים וסיכומי נתונים. Hive מאפשרת ביצוע שאילתות SQL מופשטות ישירות מקוד הjava. Hive בנויה ע"ג Hadoop(framework לעיבוד big data באופן מבוזר- ע"ג אלפי nodes[יחידות עיבוד] המטפלות במידע בהיקף של פטה-בתים בו זמנית). [↑](#footnote-ref-2)