**ארכיטקטורת הפרויקט:**

הפרויקט מכיל שלושה רכיבים עיקריים: שרת (Server), לקוח משדר (TC) ולקוח מקבל (RC).

לכל אחד מהרכיבים יש חלק משמעותי ושונה בפרויקט

**תיאור הארכיטקטורה המוצעת:**

שרת (Server):

השרת אחראי על העברת המידע בין הלקוח המשדר (TC) ללקוח המקבל (RC).

נוסף על כך השרת אחראי על פעולת התמלול והכתיבה למסד הנתונים.

השרת מתחלק לשלושה חלקים: TC\_Handler, RC\_Handler & Transcriptor.

כל אחד מהחלקים אחראי על אחת מהפעולות הראשיות של השרת (קבלה מלקוח משדר, תמלול וסכרון הוידאו והטקסט, העברה ללקוח המקבל).

TC\_Handler:

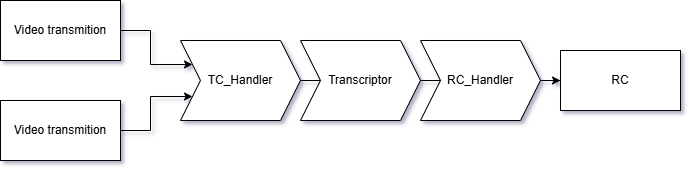
חלק זה בשרת אחראי על קבלת וידאו ואודיו מהלקוח המקבל בעזרת חיבור UDP עם פרוטוקול RTP ומסנכרן ביניהם לפני שהוא מעביר לרכיב השרת הבא.

Transcriptor:

חלק זה לוקח את האודיו ובעזרת קריאת API במטרה לתמלל את האודיו אשר התקבל מה-TC

RC\_Handler:

חלק זה אחראי על שליחת הוידאו והתמלול ל-RC בעזרת חיבור TCP, HTTP, RTP ו-WebSockets.



לקוח משדר (TC):

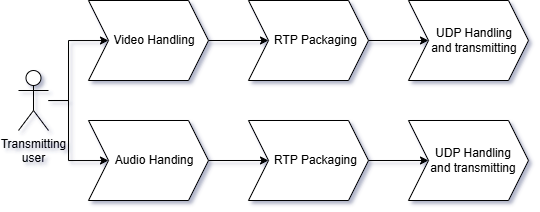
הלקוח המשדר אחראי על שידור הוידאו והאודיו לשרת.

רכיב זה הוא תוכנה לוקלית שמשדרת בעזרת חיבור UDP ופרוטוקול ה-RTP אודיו ווידאו בשני תהליכים נפרדים (פורט לוידאו ופורט לאודיו)

רכיב זה מכיל שני חלקים: Audio\_Handler, Video\_Handler.

כל חלק מהרכיב בנוי משלושה שלבים עיקריים: קליטה - הכנסה לפקטה – שליחה.

לדוגמא: קליטה של וידאו -> הכנסה לפקטת RTP -> שליחה לשרת.



לקוח מקבל (RC):

תפקידו של רכיב ה-RC הוא הצגת שידורי וידאו ותמלול המתקבלים באופן שוטף מהשרת.

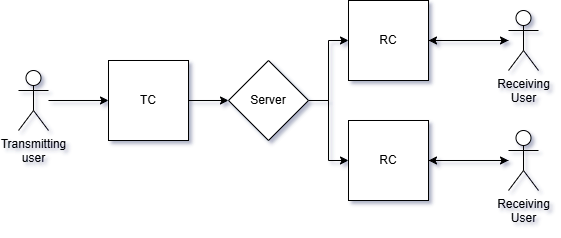
יישום זה מבוסס ווב (Web-based) ומנצל את פרוטוקול WebSocket להעברת נתונים בצורה מהירה ויעילה, המבטיחה תקשורת בזמן אמת (RTC) חיונית לשידורים חיים.

היתרונות בבחירת ארכיטקטורת ווב עבור ה-RC:

* נגישות גבוהה: המערכת נגישה מכל מכשיר המצויד בדפדפן אינטרנט.
* פיתוח ופריסה יעילים: קלות המימוש והתחזוקה בסביבת ווב.
* מתאים למגון רחב של פלטפורמות (cross-platform): עבודה חלקה על פני מערכות הפעלה ודפדפנים שונים ללא צורך בהתאמות ספציפיות.
* חווית משתמש נוחה: ממשק מוכר ואינטואיטיבי המקל על השימוש.

מאחר ומדובר באתר כל הקוד של ה-RC מאוחסן בתוך ה-Server אך ניגשים אל

ה-RC דרך הדפדפן.

כל שלושת הרכיבים הללו ביחד מנהלים את המוצר הסופי והכרחיים לתפעולו היעיל. 

**תיאור הטכנולוגיה הרלוונטית:**

שפות תכנות: השפות השונות המשומשות בפרויקט הזה.

|  |  |  |
| --- | --- | --- |
| שם השפה | שימוש | סיבה לשימוש |
| Python | Server, RC, TC | שפה ראשית של הפרויקט.  נוחה לשימוש, כתיבה מהירה, מכילה ספריות רבות ומאוד ורסטילית. מוכרת במימוש הנוח שלה לרשתות |
| TypeScript | RC | משומשת למימוש ה-RC.  השפה משומשת בצורה נרחבת בעולם פיתוח הווב. נותנת את כל מה שיש ב-JS אבל עם בטחון של OOP ושימוש ב-Types. |
| HTML5 & CSS | RC | שפות ידועות לפיתוח ועיצוב אתרים. |
| SQL | Server | שפה ידועה ובטוחה לכתיבת מסדי נתונים. |

מערכות הפעלה: מערכות ההפעלה המשומשות בפרויקט

|  |  |  |
| --- | --- | --- |
| שם המערכת | שימוש | סיבה לשימוש |
| Windows | Server, RC, TC | מערכת הפעלה ידועה, נוחה לשימוש, ורסטילית ושומשה לפיתוח ובדיקות. |

ניתן להריץ את הפרויקט על כל מערכת הפעלה כל עוד מותקנים התוספים הרלוונטים

פרוטוקולים: הפרוטוקולים המשומשים בפרויקט

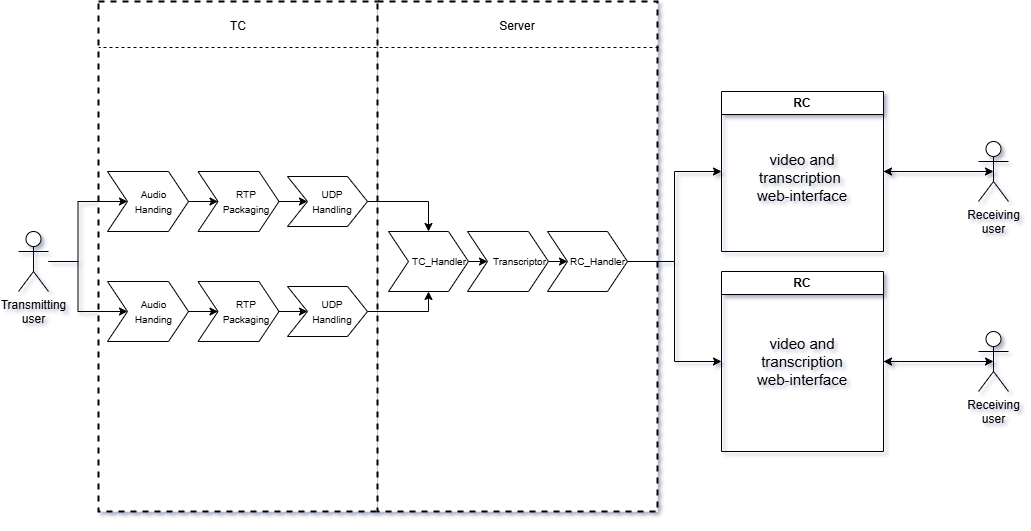
|  |  |  |
| --- | --- | --- |
| שם הפרוטוקול | שימוש | סיבה לשימוש |
| RTP | Server, RC, TC | פרוטוקול ידוע בתעשייה, נוח לשימוש וורסטילי |
| HTTP | Server, RC | פרוטוקול בו משתמשים בשביל לתקשר עם ה-browser |
| Websocket | Server, RC | פרוטוקול מעל HTTP שנותן לשרת יכולת לשלוח לקוח ווב הודעות ללא בקשה של הלקוח. |
| TCP | Server, RC | פרוטוקול אמין, מוודא הגעת הודעות, חובה לשימוש ב-HTTP |
| UDP | Server, TC | פרוטוקול מהיר יותר וידוע בשימוש בשביל סטרימינג |

תחומי עניין רלוונטים: שונות ששומשו בפרויקט.

**???????**

**תיאור זרימת המידע במערכת:**

להלן תרשים המייצג את התהליך של המערכת



**תיאור האלגוריתמים המרכזיים:**

**???????**

**תיאור סביבת הפיתוח:**

סביבות פיתוח (IDE):

במהלך פיתוח פרויקט זה היה שימוש בשתי סביבות פיתוח:

* Pycharm
* Visual Studio Code

כל סביבה שומשה למטרה ספציפית בה היא מוכרת כהכי נוחה ומכילה כלים העוזרים בפיתוח.

|  |  |  |
| --- | --- | --- |
| סביבת הפיתוח | שימוש | סיבת הבחירה |
| Pycharm | פיתוח כל החלקים בפרויקט הדורשים הכתובים ב-Python  (TC & Server) | Pycharm היא סביבת הפיתוח הכי ידועה בשביל Python ומכילה כלים רבים (כגון דיבאגר מצוין) ואינטגרציות רבות (כגון גיט)  נוסף על כך אני משתמש ב-Pycharm שנים רבות ויודע איך לתפעל את התוכנה. |
| Visual Studio Code | פיתוח ה-RC. | VSC היא תוכנת עריכת קוד אשר משומשת רבות לפיתוח של אתרים וכתיבת תוכנה לווב. היא מכילה אינטגרציות של צפייה באתר ב"לייב" בנוסף לעוד כל מיני פיצ'רים נוספים העוזרים בבניית אפליקציות וובץ  התוכנה הזאת היא אחת התוכנות הראשונות שאני כתבתי בהן קוד ואני מכיר אותה היטב, דבר שעזר לי לבחור אותה בשביל הפרויקט הזה |

מערכת ניהול גרסאות:

בפרויקט זה כמו בכל פרויקט שלי בין אם הוא פרויקט תוכנתי או לא אני השתמשתי ב-git בשביל לשמור את השינויים שלי ולוודא שלא יקרה מצב שמשהו מפסיק לעבוד בפרויקט ואני לא יכול לחזור אחורה.

נוסף על כך על מנת שימוש נוח בגיט וצפייה נוחה בשינויים השתמשתי ב-github התור פלטפורמת ה-git שלי מאחר וזאת הפלטפורמה שאני מכיר הכי טוב.

ספריות ומודולים ששומשו בפרויקט:

|  |  |  |
| --- | --- | --- |
| שם המודול / ספרייה | סיבת השימוש | סיבת הבחירה |
| socket | ניהול תקשורת | ספרייה שמובנת בתוך Python, נוחה לשימוש, אני מכיר אותה היטב וידועה לניהול תקשורת. |
| os | אינטראקציה עם מערכת ההפעלה | מאפשר גישה לשירותי מערכת ההפעלה שמשומשים בפרויקט. |
| logging | ניהול ורישום לוגים | כלי מובנה לניהול לוגים בצורה מסודרת, גמיש וניתן להתאמה אישית בקלות. |
| time | ניהול זמן ושעון (fps) | מספק פונקציות לעבודה עם זמן, כולל מדידת ביצועים, השהיות, וקבלת תאריך נוכחי. |
| select | ניהול לקוחות מרובים  ב-RC | מאפשר ניהול לקוחות מרובים בשרתים בצורה פשוטה ונוחה |
| re | עיבוד ביטויים רגולריים | משמש לחיפוש, התאמה ומניפולציה של טקסטים עם ביטויים רגולריים, תומך בגמישות מירבית. |
| struct | עבודה עם מבני נתונים בינאריים | מאפשר המרה של נתונים בין פורמט בינארי לפורמט ש-Python מבין, חיוני בתקשורת והמרת נתונים. |
| zlib | דחיסת נתונים של וידאו ואודיו ב-TC | נוחה ופשוטה לשימוש ובנויה בתוך python |
| numpy | עיבוד וידאו | ספרייה נוחה ומוכרת שנותנת תמיכה  ב-opencv. |
| cv2 | משומשת לקליטת וידאו מה-TC ולקריאת הוידאו ב-Server | מוכרת בעולם, נוחה לשימוש ויש לי ניסיון רב איתה מפרויקטים קודמים |

|  |  |  |
| --- | --- | --- |
| שם המודול / ספרייה | סיבת השימוש | סיבת הבחירה |
| pyaudio | קליטת אודיו מה-TC | מאפשר ניהול והקלטה של אודיו, מתאים לעיבוד קובצי קול ולזיהוי דיבור. |
| math | פונקציות מתמטיות מתקדמות | מובנה בתוך python ונוח לשימוש. |
| random | יצירת מספרים רנדומליים ssrc (מספר מזהה בפרוטוקול RTP) | מובנה בתוך python, נוח וקל לשימוש. |
| thread | ניהול Threads ב-Server | מאפשר יצירה ועובדה של תהליכים בתוך התוכנה. |
| multiprocess | ניהול תהליכים ב-TC שליחת אודיו ושליחת וידאו | מספק תמיכה לריבוי תהליכים, משפר ביצועים במערכות עם עיבוד כבד ע"י הפעלת מספר ליבות של המעבד. |