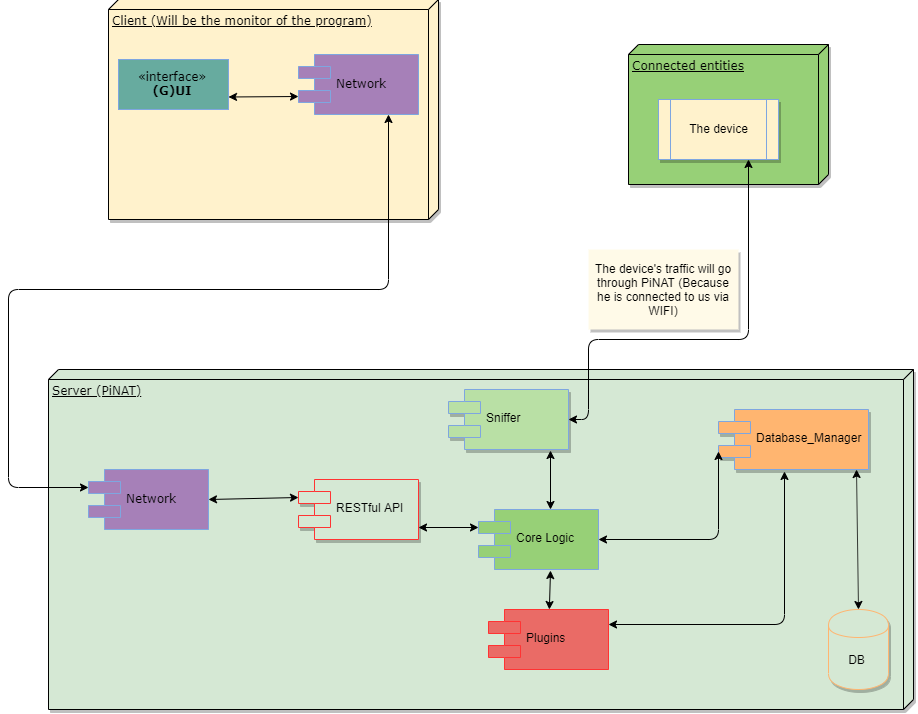
**רשימת ה-Modules:**

1. UI: בעזרת ה- RESTful API נוכל לממש בקלות ממשקי אינטראקציה למשתמש. מכיוון שהפרויקט שלנו מתמקד בעיקר בזיהוי מתקפות והפיכה של התוכנה ליעילה ומודולרית כמה שיותר, לא נתמקד ב-UI וכנראה הוא יהיה ממשק משתמש בתצורת CLI. באותה קלות נוכל לפתח אתר אינטרנט, אפליקציה ל-Mobile וכדומה.
2. Network: המודול הזה יקשיב לבקשות התחברות. ברגע של קבלה - הוא ישלח את הבקשה ל-Core Logic לעיבוד ואת התשובה יעביר חזרה למבקש. (טכנולוגיה – Python)
3. RESTful API: נוכל לבנות ממשקי משתמש שונים בעזרת ה-API הזה: הוא יוכל לקבל בקשה, לדוגמה "רשימת הפעולות שניתן לעשות עם פלאגין X" - תוך כדי תקשורת ישירה עם ה-Core Logic הוא יקבל את המידע ויחזיר אותו. מכיוון שהמידע יחזור כפורמט קבוע אך עם ערכים שונים, הממשק משתמש ידע לפרסר אותו ולהציג אותו בצורה טובה. (טכנולוגיה - Python).
4. Core Logic: הלב של PiNAT. מקבל פאקטות ובעזרת פלאגינים שונים הוא ינתח וישנה את המידע על פי הצורך ויחזיר את הפאקטות חזרה. (טכנולוגיה - Python)
5. Plugins: תהיה מערכת שתדע לטעון פלאגינים שנכתבו עם API ספציפי שנספק. הפלאגינים האלה ינתחו, אחד אחרי השני, את המידע. הם יוכלו לשנות אותו, לשמור מידע וסטטיסטיקה עליו (#8) או פשוט לא להעביר אותו הלאה. (טכנולוגיה - כנראה Python).
6. Sniffer: הרכיב שמסניף תעבורה על PiNAT, מעביר אותה ל-Core Logic ומחזיר אותה חזרה. (טכנולוגיה - Python).
7. Database\_Manager: אחראי על שמירה ושליפה של מידע ממסד נתונים **כלשהו**. פלאגינים, וגם Core Logic יצטרכו באופן שוטף לשמור מידע ולכן חיוני שרכיב זה ידע לעבוד עם מספר בקשות במקביל, וגם יהיה מודולרי אם נרצה להחליף את המסד נתונים. (טכנולוגיה - Python).
8. DB: מסד הנתונים במקרה שלנו יהיה SQLite. (טכנולוגיה – SQLite).

**דיאגרמת של כל ה-Modules:**

קישור: <https://drive.google.com/file/d/1z7RPieKUNebzPWrzo3ZNfKnrz9610XzQ/view?usp=sharing>