

מכון טכנולוגי חולון  
**הפקולטה לניהול טכנולוגית**

**בחינת שימוש בשיטות ובכליים  
למידול מערכתי**

**Evaluation of System  
Modeling Tools and Methods**

עבודת גמר לשם מילוי חלקו של הדרישות לקבלת תואר מוסמך במדעים

מאת:

אבי זגורין – 308153097

ליאור מנטיין – 300995271

בנהנחיות:

ד"ר מיכאל ויינוקור

אלון מודעי

## תודות

ברצוננו להודות בראש ובראשונה למנהלינו ד"ר מיכאל יינוקור ומר אלון מודעי על ההנחייה והליויי המקצועני, התמיכה והעזרה הרבה לאורך כל שלבי המחקר. תשומת הלב והקפדה על כל החלקים אפשרו לנו להתמקד ולהגיע לפתרונות אינטגרטיבים ולהביא את עבודת הממחקר לרמתה כיום.

כמו כן, תודה רבה לחברת Systematics ובמיוחד לאביאסף מוז שער, תמק, ליווה והכוון אותנו לאורך כל העבודה עם הכליל System Composer בצורה מקצועית וברמה גבוהה מאוד ועל כל העצות הטובות שעוזרו לנו לשפר את העבודה ולהכיר את הכליל על הצד הטוב ביותר.

## תקציר מנהלים:

על מנת לקבל החלטות בתחום הנדסת מערכות קיימים מספר כלים ושיטות שונות אשר עוזרות לمهندס המערכת לקבל את ההחלטה וזאת ע"י ביצוע מודלים, סימולציות מרובות וניתוח תוצאותיהם.

כיום מהנדסי מערכת משתמשים במידול ברמת הדיסציפלינות ההנדסית השונות ו"mdlgs" במקרים ובאים על במידול ברמת הול של הנדסת המערכת תוך שימוש בכלים במידול וشرطוט משולבים אשר אינם תלויים אחד בשני ולאו דווקא מתמשכים אחד אל השני, מצב זה מקשה על ביצוע הסימולציות וגורם לمهندס המערכת לא מעט פעמים להשתפק בשרטוט בלבד או בשילובו עם כלי אחר, בעל יכולות חסרות כגון:

- הגדרה פורמלית ומלאה של כל אלמנט
- תיאורים ומידע נוספים (על מה שימוש בתרשימים)
- קשרים בין אלמנטים
- שמרות חוקיות התרשימים
- ויזוא נכונות ושלמות
- יכולת הרצה (סימולציה) לצורך אימונות

רצינול וענין עבדות מחקר זו הם ההשוואה בין מספר שפות שונות ושיטות במידול שונות והחלה מהו השילוב היעיל ביותר מהנדס המערכת וזאת על פי מדדים לבחינת שאלת המחקר לצרכי הנדסת מערכות.

בנוסף, נבחן עד איזו רמה כדאי להעמיק בבניית המודלים וביצוע הסימולציות בפרויקט וזאת ע"י ניסויים מעשיים. כמו כן, במהלך המחקר התבכע לימוד עמוק לצורך במידול מערכות באמצעות הכלים והשיטות השונות שנבחרו. כאשר המחקר כולל שלושה חלקים עיקריים:

1. בוחינת שיטות וכלים על בסיס המידע בספרות הטכנית.
2. לימוד עמוק של הכלים, במידול המערכות וסימולציה באמצעות הכלים והשיטות שבחרנו למחקר ההשוואתי המבוסס על ניסוי מעשי עם כלים ודוגמאות.
3. הצגת תוצאות המדדים שנבחרו מענה לשאלות המחקר, על בסיס הפעולות המعيشית של סעיף 2 לעיל.

יש לציין שמרחב השיטות והכליים הקיימים רב מאוד ולכן נדרש סינון ראשוני של אותן השיטות והכליים על מנת לשמר על היקף מציאותי של העבודה המחקר ולא פחות חשוב מזה להביא תוצאות שימושיות למחנדי מערכות, תוצאות אשר יוכלו להשתמש ולישם באמצעות עבודה זו ובפרסומים הצפויים בספרות כתוצאה ממנה. בהתאם לכך נבחרו כלים בשיטות שנמצאים בשימוש כמפורט לעיל, במספר לא מבוטל בארגוני פיתוח בארץ.

Modeling tools: System Composer and Enterprise Architect or Rhapsody

Modeling languages: ECSAM and SysML

Diagram tools: Visio

בפרויקט שלנו הקשרים בין שיטות/שפות כל המודול וכלי השירותים הם:

System Composer with ECSAM .1

Enterprise Architect with SysML .2

Visio with ECSAM .3

במחקר זה אנו נבחן ונציג את יתרונות הכלים (הנבחרים) השונים בשילוב שפות המודול (הנבחרות) השונות וזאת באמצעות 2 דוגמאות ו-6 מדדים אשר יעוזו לנו לבחון ולענות על שאלת המחקר הראשית ושאלות המחקר המשניות.

שאלות המחקר הראשית של פרויקט גמר זה הן:

1. מהי שיטת המודול האידיאלית עבור הנדסת מערכות ובאיזה שכבה יש לבצעה?
2. מה יעיל יותר עבור מהנדס המערכת, כל מודול מלוחה בשפה/שיטה ידועה בעל יכולת קישור אל הפיזיקה של חלקי המערכת בתוך אותו כל או שילוב של כל מודול וشرطויות מערכתיים שונים?

המדדים שנבחרו לבחינת שאלות המחקר הם:

נוחיות ויזואלית

ביצוע הגדרת חוזרת / עדכון הגדרה

IN/OUT – שינויים כניסה ויציאה מהמערכות השונות

שינויי ארכיטקטורה באמצעות הזזת אלמנטים

שפה מודול מוכרת לכל המשתתפים בפרויקט

השפעה של שינויים באמצעות הפרויקט

על מנת לבדוק את שאלות המחקר נבחרו 2 דוגמאות שמאפייניות את בעיות ה"קיזון" שמאתגרות כבר מספר שנים את עולם המידול.

**1. "ניטור פגיעה ברכב 360" - מערכת משולבת עם סיבוכיות גבוהה בעלת תתי מערכות עם**

**סיבוכיות ביוננית/גובה:**

בדוגמה זו נבחן רכיב אחד מאוד מורכב, אשר מורכב ממספר חישנים שונים.

הרכיב אשר נבחן הוא גוף הרכב אשר אליו מחוברים חישנים שונים המדווחים אל מערכת הזיהוי פגיעיות המדוחחת אל מרכז הבקרה.

המחקר בעצם בוחן את הכלים השונים ברמת הקושי לבצע מידול של מערכת אחת מורכבת בעלת תתי מערכות רבות.

דבר זה מסבך מאוד את בניית הסימולציה עצמה אשר מצריכה תיאור מלאה של התנהוגיות המערכת, ירידה לפרטים, בנייה של כל תת מערכת וההתנהוגיות השונות שלה וביצוע הסימולציה על כל תת מערכת בנפרד ועל המערכת כולה בשלמותה.

**2. "ניהול מערך קרונות חכם" - מערכת משולבת עם סיבוכיות ביוננית/גובה בעלת הרבה**

**תתי מערכות זהות עם סיבוכיות נמוכה:**

בתרחיש זה נבחן מספר רב של רכיבים פשוטים המדוחחים נתונים אל מרכז הבקרה אשר מהווה את מרכז העיבוד והחישוב, למעשה הוא זה שמקבל את החלטות.

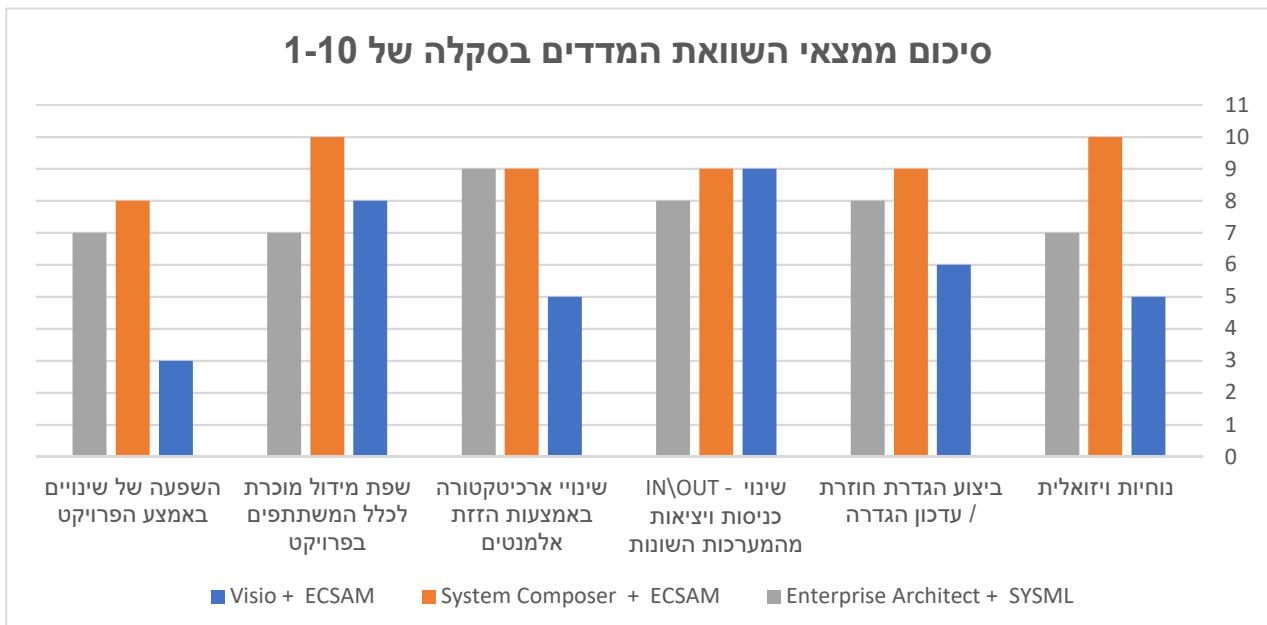
הרכיבים אשר נבחנו הם קרונות תחבורה אוטונומיים המדוחחים נתונים אל מרכז בקרה חכם אשר מעבד את הנתונים מכל הקרונות ומבצע את ההחלטה.

המחקר בעצם בוחן את הכלים השונים ברמת הקושי לבצע שכפול רכיבים והתמודדות עם כל המידע המגיע מכל רכיב בנפרד לאחר השכפול, יוצרה של רכיב אחד פשוט והכפלתו ב- $n$  פעמים.

דבר זה משפיע מאוד על בניית הסימולציה עצמה אשר מצריכה תיאור מלאה של התנהוגיות המערכת, ירידה לפרטים, בנייה של תת המערכת והכפלתה ב- $n$  פעמים וביצוע הסימולציה על המערכת כולה בשלמותה.

במקרה שלנו, תחילת בינוי את המערכת לפי קרון אחד וכשרצינו להכפיל אותו גילנו שיש שינוי גדול בין קרון 1 לבין 2 קרונות ומעלה, דבר אשר השפיע וגרם לכך שארכיטקטורת המערכת ברמה הלוגית השתנתה.

סיכום תוצאות המדרדים עברו שלושת הכלים שנבחרו עבור כל מדריך נפרד מוצגים בגרף הבא:



להלן טבלה המסכםת את תוצאות המדרדים המשוקלים עבור כל כלי + שיטה בנפרד:

תוצאות סיכום מדרדים משוקל	כלי ושיטה
9	System Composer + ECSAM
7.5	Enterprise Architect + SysML
5.3	Visio + ECSAM

### המסקנות העיקריות שלנו מהפרויקט הנ"ז:

ישנה עדיפות רבה שלשלב הشرطוטים ה"חופשיים" יבנה בשיקול דעת וכובד ראש ובן עט זאת, לא לאפשר לחלק זהה לגוזל זמן יקר מהפרויקט, שכןشرطוטים הללו ישתנו מספר פעומים, ואם יבנו בכללי מידול אשר כולל את הסימולציה הشرطוטים יבנו מחדש בכל מקרה. בהתאם לכך סעיף זה מתיחס לפרויקטים קטנים וגדולים כאחד וזאת בזכות יתרונותיו הרבים של כל הميدול השונים (כגון: נוחות ויזואלית, פרמטרים ואפשרויות רבים, יכולות הפekt דוחות ועוד).

עובדת עם כל הميدול בודד מתחילה הפרויקט ועד שלב הסימולציה מאפשרת ייעול של העבודה ללא צורך בהתמכחות בין כלים שונים, שפות שונות ועוד. כמו כן, עבודה עם כל הميدול שיעוד לבצע גם אתشرطוטים עצם ייעיל ויחסוך זמן יקר בשרטוט כפול של תרשימי המודלים של הפרויקט.

העבודה עם כל הميدול אשר מקנה לנו יכולת הפekt דוחות ובנית ארכיטקטורה ברמה גבוהה עוזר מאוד ומיעיל מאוד את התהליך, נותן לנו לראות בבירור את כל הקישורים השונים, בכל ההיררכיות השונים, תוך שימוש בפילטרים לפי הצורך, דבר אשר מפשט מאוד את ההסתכלות עלشرطוטים.

יש לבצע כמה שיטות סימולציות, כמה שיתר מקיפות, אם ישנה אפשרות לאוטומציה, רצוי מאוד להשתמש בה, בפרויקט שלנו ביצענו בדיקות ידניות ואוטומטיות.

בכל הבדיקה System Composer, ישנו יכולות הקלטה לצורך חקירה לאחר, עקב גודל המחקר והזמן המוגבל תוכנה זאת רק הדוגמה לנו ולא נבחנה על ידנו אך ניראת בתוכנה חשובה ושימושית.

### הלקחים העיקריים שלנו בפרויקט הם:

ביצוע סימולציות רבות ככל האפשר לפני תחילת פיתוח הפרויקט האמיתי, פעולה אשר יכולה להתגלות כחיסכון עצום בעלות, זמן ועוד.

לא להסתמך על "شرطוטים חופשיים" בלבד ולהתחל את שלב בנית המודלים כמה שיתר מוקדם.

הפekt דוחות בכל שלב בפרויקט על מנת להבחן בעיות השונות כמה שיתר מוקדם, דבר אשר יכולות לחסוך זמן ומשאבים רבים.

### מגבליות המחקר עבור פרויקט מחקר זה הן:

זמן – עקב הזמן המוגבל, כמות הכלים הנחקרים הייתה גם היא מוגבלת ולכן בחרנו להתייחס ל-2 כלים אשר בשימוש נרחב בתעשייה עם השיטות המתאימות כפי שתואר בפרק 1.

קישור המודל אל כלל המערכת – קישור 2 הדוגמאות השונות באמצעות SC אל הפיזיקה של המערכות (כגון: גוף הרכב, גוף הקرون ועוד).

לימוד הכלי System Composer אשר נמצא בשלבי פיתוח ושינויים רציפים היו עבורנו מכשול, הוא בהיבט התיאורטי והן בהיבט המעשי.

קורונה – בעיצומו של המחקר החלה מחלת הקורונה והטגר, עקב כך, ההנחיה על הכלי System Composer, כדי מרכיב מאד, ללא אינטראקציה אישית הקשתה את תהליך הלמידה.

כיווני המשך העיקריים לפרויקט מחקר זה הם:

המשך הפרויקט וקישרוו אל החלק הפיזי של הממערכות.

הוספת חיישנים וחלקים נוספים.

ניסוון ביצוע פעולות אוטומטית באמצעות Scripts הן לSIMOLIZER מערכת והן להפקת דוחות.

ביצוע סימולציות בכלים נוספים כגון Enterprise Architect והשוואה אל המחקר הנוכחי.

השוואת כל מידול/شرطוט נוספים מקוד פתוח (כגון: ArgoUML, BOUML) והשוואה אל כל המידול/شرطוט במחקר זה. כמו כן, לדון בעניין בטחון המידע בשימוש בכל מידול/شرطוט קוד פתוח.

## תוכן עניינים

20.....	<b>פרק 1 – מבוא</b>	1.
20.....	<b>רציון</b>	.1.1
22.....	<b>חשיבות המחבר</b>	.1.2
22.....	<b>דוגמאות לפרויקטים אשר יבחן במהלך המחבר:</b>	.1.3
22.....	<b>מטרת המחבר</b>	.1.4
23.....	<b>שאלת מחקר ראשית</b>	.1.5
23.....	<b>שאלת מחקר משנהות</b>	.1.6
23.....	<b>מדדים לבחינת שאלת המחבר לצרכי הנדסת מערכת</b>	.1.7
24.....	<b>תורחיש מערכות</b>	.1.8
26.....	<b>פרק 2 – סקר ספורות</b>	2.
26.....	<b>מבוא לפך</b>	.2.1
27.....	<b>שיטות וכלי מידול</b>	.2.2
28.....	<b>שפות מידול</b>	.2.3
28.....	<b>שפה המידול Systems Modeling Language</b>	.2.3.1
32.....	<b>Embedded computer system analysis and modeling</b>	.2.3.2
33.....	<b>כלי מידול</b>	.2.4
34.....	<b>כלי המידול בשילוב שפות המידול</b>	.2.5
35.....	<b>מצב היישום בתעשייה בנושא סימולציה</b>	.2.6
38.....	<b>סקירת החישנים</b>	.2.7
40.....	<b>סיכום הפרק</b>	.2.8
41.....	<b>פרק 3 – מתודולוגיה</b>	.3
41.....	<b>הפרדיגמה המחקרית</b>	.3.1
46.....	<b>מדגם המידול והסימולציה</b>	.3.2
46.....	<b>ניטור פגעה ברכיב 360</b>	.3.2.1
48.....	<b>ניהול מערך קرونוט חכם</b>	.3.2.2
50.....	<b>טבלת תrhoחישים והסביר מדוע חישנים</b>	.3.2.3
59.....	<b>מודול וסימולציה באמצעות System Composer</b>	.3.2.4
64.....	<b>מודול באמצעות Enterprise Architect</b>	.3.2.5
69.....	<b>עץ צרכים</b>	.3.3
70.....	<b>טבלת דרישות</b>	.3.4
70.....	<b>טבלת דרישות - ניטור פגעה ברכיב 360</b>	.3.4.1
74.....	<b>טבלת דרישות - ניהול מערך קرونוט חכם</b>	.3.4.2
79.....	<b>פרק 4 – תוצאות המחבר – בדיקת مدى המחבר</b>	.4

81	.....	<b>System Composer</b>	.4.1
92	.....	<b>Enterprise Architect</b>	.4.2
103	.....	<b>Visio</b>	.4.3
114	.....	<b>סיכום ממצאי השוואת המדדים בסקלה של 1-10 (1=לא טוב, 10=מצוין)</b>	.4.4
116	.....	<b>יכולת אימות הניתוח על ידי סימולציה</b>	.4.5
121	.....	<b>פרק 5 – סיכום</b>	.5
121	.....	<b>מסקנות עיקריות מהפרויקט</b>	.5.1
123	.....	<b>לקחים עיקריים</b>	.5.2
124	.....	<b>מוגבלות המחקר</b>	.5.3
125	.....	<b>כיווני המשך מומלצים</b>	.5.4
126	.....	<b>רשימת מקורות</b>	
131	.....	<b>נספחים</b>	
131	.....	<b>נספח א: סקירת החישנים (מתיחס לסעיף 2.5)</b>	
136	.....	<b>נספח ב: תיאור התרחישים והמופעים (מתיחס לסעיף 3.2)</b>	
184	.....	<b>נספח ג: תיאור המידול והסימולציה - (מתיחס לסעיף 3.2)</b>	
205	.....	<b>נספח ד: אימות משקלים המדדים (מתיחס לסעיף 1.7)</b>	

## חלוקת אחראיות בפרויקט

בפרויקט זהחלוקת האחראיות הייתה לפי כלי המידול אשר נבחנו.

1. אבי – אחראי על תקצير המנהלים, פרק המבוא וכלי המידול והסימולציה (בניתן הארכיטקטורה, מודלים, state flow, System Composer בפועל ועוז) וכלי השירות Visio.
2. ליאור – אחראי על סקר הספרות וכלי המידול Enterprise Architect (בניתן הארכיטקטורה BDD, IBD, Sequence Diagram, Use Case Diagram ועוז)

כמו כן, הספר נכתב ונבנה ע"י 2 השותפים לפרויקט (אבי וליאור), כל אחד בתחום אחראותו (כלי המידול שבאחריותו) עבור הפרקים: מתודולוגיה, תוכאות המחקר וסיכום.

## רשימת איורים

עמוד	שם + מספר	פרק
28	איור 2.1: סקירה כללית של קשרי גומלין בין SysML / UML	2.3.1
29	איור 2.2: דוגמה ל- BDD מכלי מידול Enterprise Architect	2.3.1
30	איור 2.3: דוגמה ל- IBD של בלוק במערכת מכלי מידול Enterprise Architect	2.3.1
31	איור 2.4: דוגמה ל- IBD של מערכת מכלי מידול Enterprise Architect	2.3.1
59	איור 3.1 - ארכיטקטורה כללית System Composer	3.2.4
60	איור 3.2 - מודל ארכיטקטורת החישנים System Composer	3.2.4
60	איור 3.3 - סכימת חישון GPS System Composer	3.2.4
60	איור 3.4 - state flow של חישון בודד System Composer	3.2.4
62	איור 3.5 - סימולציה - מצב ההתחלתי System Composer	3.2.4
62	איור 3.6 - סימולציה - נתונים זרים System Composer	3.2.4
63	איור 3.7 - סימולציה - שינוי נתונים System Composer	3.2.4
63	איור 3.8 - סימולציה - מעבר באמצעות התניתה System Composer	3.2.4
64	איור 3.9 - UCD (Use Case Diagrams) - Enterprise Architect	3.2.5
65	איור 3.10 - SD (Sequence Diagrams) - Enterprise Architect	3.2.5
65	איור 3.11 - BDD של כלל המערכת - Enterprise Architect	3.2.5
66	איור 3.12 - IBD של בלוק בודד - Enterprise Architect	3.2.5
67	איור 3.13 - IBD של כלל המערכת - Enterprise Architect	3.2.5
68	איור 3.14 - סרגל התמצאות Enterprise Architect	3.2.5

69	איור 3.15: עץ צרכים	3.3
81	איור 4.1 – תפריט ניתוב בין רכיבים בארכיטקטורה כללית System Composer :4.1	4.1.1
82	איור 4.2 State flow – System Composer :4.2 בתוכו רכיב	4.1.1
83	איור 4.3 – הציגת קישורי הרכיב System Composer :4.3	4.1.1
83	איור 4.4 – ציון עברו מודד "נוחיות ויזואלית" System Composer :4.4	4.1.1
84	איור 4.5 – ציון עברו מודד "ביצוע הגדרת חוזרת / עדכון הגדרה" System Composer :4.5	4.1.2
85	איור 4.6 – בחירת סוג ומצב כניסה ויציאה 1 System Composer :4.6	4.1.3
86	איור 4.7 – בחירת סוג ומצב כניסה ויציאה 2 System Composer :4.7	4.1.3
86	איור 4.8 – ציון עברו מודד "IN\OUT - שינוי כניסה ויציאות מהמערכות השונות" System Composer :4.8	4.1.3
87	איור 4.9 – בחירת רכיב System Composer :4.9	4.1.4
87	איור 4.10 – ציון עברו מודד " שינוי ארכיטקטורה באמצעות הזוזת אלמנטים" System Composer :4.10	4.1.4
88	איור 4.11 – ספירה של קלילים לסימולציה System Composer :4.11	4.1.5
89	איור 4.12 – ציון עברו מודד " שפת מידול מוכרת לכל המשתתפים בפרויקט" System Composer :4.12	4.1.5
90	איור 4.13 – ציון עברו מודד " השפעה של שינויים באמצעות הפרויקט" System Composer :4.13	4.1.6
91	איור 4.14 – גרפ' ציוני מדדים System Composer + ECSAM :4.14	4.1
92	איור 4.15 – סרגל התמצאות Enterprise Architect :4.15	4.2.1
93	איור 4.16 – IBD של כלל המערכת Enterprise Architect :4.16	4.2.1

93	איור 93 – ציון עבור מודד "נוחיות ויזואלית" Enterprise Architect :4.17	4.2.1
95	איור 95 – ציון עבור מודד "ביצוע הגדרת חוזרת / עדכון הגדרה" Enterprise Architect :4.18	4.2.2
96	איור 96 – הוספת פורט ברכיב קיים Enterprise Architect :4.19	4.2.3
96	איור 96 – ציון עבור מודד "IN\OUT - שינוי כניסה ויציאה מהמערכות השונות" Enterprise Architect :4.20	4.2.3
97	איור 97 – חישנים עצמאיים Enterprise Architect :4.21	4.2.4
98	איור 98 – חישנים בתוך בלוק יודי Enterprise Architect :4.22	4.2.4
98	איור 98 – ציון עבור מודד "שינוי ארכיטקטורה באמצעות הזנת אלמנטים" Enterprise Architect :4.23	4.2.4
99	איור 99 – מדריך המשתמש Enterprise Architect :4.24	4.2.5
100	איור 100 – ציון עבור מודד "שפת מידול מוכרת לכל המשתתפים בפרויקט" Enterprise Architect :4.25	4.2.5
101	איור 101 – ציון עבור מודד "השפעה של שינויים באמצעות הפרויקט" Architect Enterprise :4.26	4.2.6
102	איור 102 – גוף ציוני מדדים Enterprise Architect + SysML :4.27	4.2
104	איור 104 – ארכיטקטורה כללית של "ניהול מערכ קרונות חכם" Visio :4.28	4.3.1
104	איור 104 – מערכ הקרונות (קרון בודד) Visio :4.29	4.3.1
104	איור 104 – ציון עבור מודד "נוחיות ויזואלית" Visio :4.30	4.3.1
105	איור 105 – ארכיטקטורה מצב ישן Visio :4.31	4.3.2
106	איור 106 – ארכיטקטורה מצב חדש Visio :4.32	4.3.2
106	איור 106 – ציון עבור מודד "ביצוע הגדרת חוזרת / עדכון הגדרה" Visio :4.33	4.3.2

107	איור 4.34 – שינוי ציוני חצים Visio	4.3.3
108	איור 4.35 – ציון עבור מודד "ביצוע הגדרת חוזרת / עדכון הגדרה" Visio	4.3.3
109	איור 4.36 - ארכיטקטורה כללית Visio	4.3.4
110	איור 4.37 – ציון עבור מודד "ציון המודד הזוזת "שינויי ארכיטקטורה באמצעות הזוזת אלמנטים"	4.3.4
111	איור 4.38 – ציון עבור מודד " שפט מידע מוכרת לכל המשתתפים בפרויקט"	4.3.5
112	איור 4.39 – ציון עבור מודד "השפעה של שינויים באמצעות הפרויקט"	4.3.6
113	איור 4.40 – ECSAM + Visio - גרפ' ציוני מדדים	4.3
115	איור 4.41: גרפ' סיכום ציוני מדדים	4.4
117	איור 4.42 : Composer System – סימולציה – הכנסת ערכים	4.5
118	איור 4.43 - סימולציה - Flow State - Composer System	4.5

## רשימת טבלאות

עמוד	שם + מספר	פרק
23	טבלה 1.1: טבלת מדדים	1.7
43	טבלה 3.1: קורלציה בין שאלות המחקר למדדים	3.1
44	טבלה 3.2: קורלציה בין המדדים לדרך מדידתם	3.1
46	טבלה 3.3: ניטור פגיעה ברכיב 360 - טבלת תרחישים ומופעים	3.2.1
48	טבלה 3.4: ניהול מערכ קرونות חכם - טבלת תרחישים ומופעים	3.2.2
50	טבלה 3.5: ניטור פגיעה ברכיב 360 - טבלת תרחישים ומופעים עם נתוניים אמיתיים	3.2.3.1
53	טבלה 3.6: ניטור פגיעה ברכיב 360 - טבלת ערכי חישננים	3.2.3.2
55	טבלה 3.7: ניהול מערכ קرونות חכם - טבלת תרחישים ומופעים עם נתוניים אמיתיים	3.2.3.3
58	טבלה 3.8: ניהול מערכ קرونות חכם - טבלת ערכי חישננים	3.2.3.4
70	טבלה 3.9: ניטור פגיעה ברכיב 360 - טבלת דרישות	3.4.1
74	טבלה 3.10: ניהול מערכ קرونות חכם - טבלת דרישות	3.4.2
79	טבלה 4.1: טבלת מדדים	4
114	טבלה 4.2: טבלת סיכום ציוני מדדים	4.4
115	טבלה 4.3: טבלת סיכום ציונים משוקלים	4.4
120	טבלה 4.4: טבלת סיכום סימולציות לפי פרויקט	4.5

## מילון מושגים

- **Verification and Validation (V&V)** - אימות ותיקוף של המערכת לעשות את הדבר הנכון, ולעשות אותו נכון.
- **SEMP** - המסמך קובע את התכנית הכלולת לפיתוח הטכני של הפרויקט. הוא מגדר את הפרמטרים לביצוע המערכת ותכורת מערכת שמספקת את הדרישות הטכניות ומאפשרת תכנון ובקרה של המשימות הטכניות בתכנית. בנוסף, הוא מספק מסגרת מקצועית לעמידה ביעדי עלות, ביצועים טכניים, איכות, ויעדי לוח הזמנים של הפרויקט או התכנית.
- **Requirements Verification Matrix (RVM)** - מסמך המגדיר את עקרונות בדיקת מכלול דרישות המערכת מגדר פירוט תהליכי האימות לאוסף דרישות המערכת ועיטוין מספק לתוכנית את עקרונות תוכנית הבדיקה, כשהוא מפרט: בדיקה, ניסוי, אנליה, בחינה או הדגמה בסקרים יוצגו אותם נושאיהם שנראה שהפרויקט לא יוכל לעמוד בהם או עבר בקרה גבולית את בדיקות האימות.
- **Software Requirements Specification (SRS)** - מסמך המתאר מה התוכנה תעשה ואיך היא צפיה לבצע. כמו כן, מסמך זה מתאר את הפונקציונליות שה מוצר צריך כדי למלא את כל בעלי העניין (העסק, המשתמשים וכו').
- **Use Case Diagram** - דיאגרמת התנהגותית אשר מורכבת מניטוח תרחישי שימוש, היא למעשה דיאגרמה התנהגותית אשר מרכיבת מניטוח תרחישי שימוש. מטרתה היא הצגת הרקע הפונקציונלי של המערכת, ה תלות בין האלמנטים הקשורים למערכת (בתוכה ובסביבה) והפעולות שבוצעות ע"י כל אחד מהם.
- **Sequence Diagram** - תרשימים רציף הנועד לתיאור והסביר התהליך כיצד המערכת ורכיביה מבצעים תהליך זה.

- **Visio** - כלי המאפשר ייצור תרשימי זרימה (דיגרמות) ויזואליים, סטטיים.
- **SysML** - שפה כללית לצורכי יישום הנדסת מערכות, בעל תמייה במפרט, ניתוח, תכנון, אימות ותיקוף של מגוון systems and systems-of-systems.
- **ECSAM** - שיטה להציג דרישות הנדסיות ומודלים למערכות מבוססות מחשב. מודל זה מציג מערכת באמצעות שלוש תצוגות (Module Chart, Activity Chart, StateChart) עיקריות ומשלים העוסקות בארQUITקטורה הריעונית, אופני הפעולה, שהם הבסיס הלוגי ל贊ינות היכולות.
- **System Composer** - כלי המאפשר הגדרה ובניה של ארQUITקטורות וקומפוזיציות להנדסת מערכות, מבוססת מודל ותוכנן תוכנה. באמצעות כלי זה ניתן להציג דרישות תוך שיפור ארQUITקטורת המודל הנחננת לתכנון באמצעות הכלים Simulink.
- **State Flow** – כלי המספק שפה גרפית הכוללת דיגרמות מעבר, תרשימי זרימה, טבלאות מעבר ועוד. באמצעות כלים אלה ניתן לתאר כיצד אלגוריטמים של MATLAB ומודלים של Simulink מבינים אותה.
- **Simulink** – כלי תכנון והדמיה (סימולציה) של מודלים אשר הוכנו מראש.
- **Enterprise Architect** - זהו כלי מידול ועיצוב המבוסס שפת UML. הפלטפורמה תומכת בתכנון ובנית מערכות תוכנה, מידול תהליכי עסקים וכו'. בנוסף לכלי מידול זה יכולות לביצוע הנדסת מערכות ע"פ שפת SysML.
- **Block Definition Diagram (BDD)** - משמש להגדרת מאפייני הישויות (הבלוקים) הפיזיות והריעוניות של המערכת, כולל תוכנותיו ומשקיו של כל ישות.
- **Internal Block Diagram (IBD)** - משמש לתיאור המבנה הפנימי של הבלוק, מצין את מאפייניו והאינטראקטיות בתוכו. בנוסף, משמש גם לציון המאפיינים והאינטראקטיות בין הבלוקים במערכת.

## מילון קיצורים

פירוש הקיצור	קיצור
System Composer	SC
Enterprise architect	EA
Sequence diagram	SD
Use Case Diagram	UCD
Internal Block Diagram	IBD
Block Definition Diagram	BDD

## 1. פרק 1 – מבוא 1.1. רצינול

אחד השאלות המעניינות והרלוונטיות בהנדסת מערכות היא תיאור מובנה ומתגבש הדרגתית במהלך הפיתוח של המערכת, במתכונת פורמאלית וקונסיסטנטית, המאפשרת חשיבה מוקדמת, חקר אפשרויות וחלופות, וידוא עמידה בדרישות ורציפות עם המימוש. כמו כן, מהוות גם בסיס לביקורת האימוט והתקוף של המערכת בתחום הנדסת הממערכות.

על מנת לקבל החלטות בתחום הנדסת הממערכות קיימים מספר כלים ושיטות שונות אשר עוזרות למהנדס המערכת לקבל את ההחלטה וזאת ע"י ביצוע מודלים וסימולציות רבותות וניתוח תוצאותיהם.

כיום מהנדסי מערכת משתמשים במודלים ברמות הדיסציפלינות ההנדסיות האחרות (לדוגמה - מודלים אלקטרוניים או מכניים של המערכת) ו"mdlgs" במקרים רבים על מידול ברמת העל של הנדסת המערכת. כמו כן, הם משתמשים בכלים מידול וشرطוט משולבים אשר אינם תלויים אחד בשני ולאו דווקא מתמשכים אחד אל השני, דבר אשר מקשה מאוד על ביצוע הסימולציות.

חוסר ההתקשרות בין הכלים הוא דבר המסביר עוד יותר את הנושא ומקשה על ביצוע הסימולציות השונות וכן ניתוחם. כמו כן, רגש במידה גבוהה מאוד לשינויים, גם קטנים ביותר.

כיום ביצוע מודלים וסימולציות אלה מבוצעות באמצעות מספר כלים באופן המשלב כל מידול וסימולציה, שיטה, וכל שרטוט.

לא מעט פעמים מהנדס המערכת משתמש בשרטוט בלבד או בשילובו עם כלי אחר, בעל יכולות חסרות כגון:

- הגדרה פורמאלית ומלאה של כל אלמנט
- תיאורים ומידע נוספים (על מה שמופיע בתרשימים)
- קשרים בין אלמנטים
- שמיירת חוקיות התרשימים
- וודוא נכונות ושלמות
- יכולת הרצה (סימולציה) לצורכי אימוט

במקרה היוטר נפוץ – תיאורים מילוליים-סיפוריים בלבד

רצינול וענין עוסקת מחקר זו הם ההשוואה בין מספר שפות שונות ושיטות מידול שונות וההשלטה מהו השימוש היעיל ביותר עבור מהנדס המערכת וזאת על פי מדדים לבחינת שאלת המחקר לצרכי הנדסת מערכות אשר מוצגים בסעיף 1.7.

בנוסף, נבחן עד איזו רמה כדאי להעמיק במבנה המודלים וביצוע הסימולציות בפרויקט וזאת ע"י ניסויים מעשיים.

כמו כן, במהלך המחקר התבכע לימוד עמוק לצורך מידול מערכות באמצעות הכלים והשיטות השונות שנבחרו.

המחקר כולל שלושה חלקים עיקריים:

1. בוחנת שיטות וכליים על בסיס המידע בספרות הטכנית
2. לימוד עמוק של הכלים, מידול המערכות וסימולציה באמצעות הכלים והשיטות שבחרנו למחקר ההשוואי המבוסס על ניסוי מעשי עם כלים ודוגמאות.
3. הצגת תוצאות המודדים שנבחרו למען לשאלות המחקר, על בסיס הפעולות המعيشית של סעיף 2 לעיל

מרחיב השיטות והכלים הקיימים רב מאוד. אי לכך נדרש סיכון ראשוני של אותן השיטות והכלים וזאת על מנת לשמור על היקף מציאותי של עבודת המחקר ולא פחות חשוב מזה להביא תוצאות שימושיות למתנדסי מערכות, תוצאות אשר יוכלו לשימוש ולשימוש באמצעות העבודה זו ובפרסומים הצפויים בספרות כתוצאה منها.

אי לכך נבחרו כלים בשיטות שנמצאים בשימוש כמפורט לעיל, במספר לא מבוטל בארגוני פיתוח בארץ.

Modeling tools: System Composer and Enterprise Architect or Rhapsody

Modeling languages: ECSAM and SysML

Diagram tools: Visio

בפרויקט שלנו הקשרים בין שיטות/שפות כל הميدול וכל הشرطוט הם:

System Composer with ECSAM .2

Enterprise Architect with SysML .3

Visio with ECSAM .4

## 2. חשיבות המחקר

במחקר זה אנו נבחן ונציג את יתרונות הכלים (הנבחרים) השונים בשילוב שפות המידול (הנבחרות) השונות וזאת באמצעות 2 דוגמאות אשר ייעזרו לנו לבחון ולענות על שאלת המחקר הראשית ושאלות המחקר המשניות.

### 3.1.3. דוגמאות לפרויקטים אשר יבחן במהלך המחקר:

1. מערכת משולבת עם סיבוכיות גבוהה בעלת תתי מערכות עם סיבוכיות

בינונית/גבוהה

לדוגמה: מערכת צילום לרכב המאפשר צילום 360 מעלות ותיעוד בזמן אמת של פריצות ופגיעות רכב כאשר הרכב כבוי.

2. מערכת משולבת עם סיבוכיות ביןונית/ גבוהה בעלת הרבה תתי מערכות זהות עם סיבוכיות נמוכה.

לדוגמה: סטאטוס כל הקرونוט ברשות התchapורה הציבורית העתידית של גוש דן.

## 1.4. מטרת המחקר

המחקר מתמקד בשילוב כלי המידול + שיטת המידול המיטבית, תוך התיחסות לסוג הפרויקט ולשכבה המערכת בה כדי לבצע את המידול מערכתי.

תכנון וביצוע מידול באמצעות שני כלי שונים

(System composer and Rhapsody or Enterprise Architect) וביוצע סימולציה

באמצעות הכלי System composer, כלי חדש שאמור לאפשר סימולציה משולבת של

כלי סימולציה פיזיקליים (MATLAB, Simulink and State flow) אשר נפוצים ביותר

במידול מרכיבים בודדים אשר שמהנדסים כן מבצעים, כפי צוין בסעיף 1.1 . סימולציה

מערכתית בעזרת כלים התומכים בו SysML הוצג במחקרים רבים, لكن לא בוצע במחקר זה (תיאור פרטי הנושא בפרק 2.3)

העבודה המעשית בוצעה כדלהלן:

1. בניית ארכיטקטורה (בעור כל דוגמה)

2. מידול המערכת באמצעות כלים המידול

3. השוואה בין כלים המידול באמצעות מדדים לבחינה

4. מימוש אלגוריתם התרחישים באמצעות Simulink & State flow (בעור כל

דוגמה) – יבוצע רק ב- System Composer

5. הרצת הסימולציה בכלים המידול – יבוצע רק ב- System Composer

### 1.5. שאלת מחקר ראשית

3. מהי שיטת המידול האידיאלית עבור הנדסת מערכת ובאיזה שכבה יש לבצע?
4. מה ייעיל יותר עבור מהנדס המערכת, כלי מידול מלאה בשפה/שיטה ידועה בעל יכולת קישור אל הפיזיקה של חלקים המערכת בתוך אותו כלי או שילוב של כל מידול וشرطוטים מערכתיים שונים?

### 1.6. שאלת מחקר משנהות

5. באיזה מידה יש יתרון למידול כלים על פני שימוש בשרטוטי Visio?
6. כאשר מבצעים סימולציה, באיזה שכבת מערכת יש לבצע את הסימולציה (באיזה רמה יש לבצע את התרחישים)
7. עד איזה שכבה יש ערך מסויף לביצוע הסימולציה?

- השאלות המשניות 4 ו-5 יבחן רק ברכי System Composer.

### 1.7. מדדים לבחינת שאלת המחקר לצרכי הנדסת מערכת

בטבלה 1.1 ניתן לראות את המדדים אל מול המשקלים שלהם

- תיקוף המשקלים בוצע באמצעות מדגם (ע"י שאלון [נספח ד]) שעבר בקרב מהנדסים ומנהלי פרויקטים בעלי ידע וניסיון בנושא.

משקל		מדד
20%	נוחיות ויוזאלית	1
20%	ביצוע הגדרת חוזרת / עדכון הגדרה	2
10%	IN/OUT – שינוי כניסה ויציאות מהמערכות השונות	3
10%	שינוי ארכיטקטורה באמצעות הזזת אלמנטים	4
10%	sett מידול מוכрат לכל המשתתפים בפרויקט	5
30%	השפעה של שינויים במהלך הפרויקט	6

טבלה 1.1: טבלת מדדים

טבלה 1.1 מסכמת את משקל כל מדד על בסיס הניסיון ההנדסי של מבצעי העבודה, המנהים ומשתתפי המדגם.

כמו כן, מدد נוסף אותו נבחן במחקר זה הינו: יכולת אימות הניתוח על ידי סימולציה - מدد זה יבחן ע"י כלי ה-System Composer בלבד.

#### 1.8. **תרחישי מערכת**

להלן התרחישים אשר משמשים כבסיס לניסויים המעשיים אשר משמשים כבסיס למחקר זה. ניסויים אלה הם ניסויי "מעבדה" של בניית המודלים וביצוע הסימולציות בעוזרת הכלים והשפות / שיטות המידול שנבחרו.

##### 1.8.1. **תרחישי ניטור פגיעה ברכיב 360:**

###### 1.8.1.1. **תרחיש נסיען פריצה על ידי פגיעה בחולון הרכב:**

- א. הרכיב חונה - כבוי
- ב. המערכת זיהתה פגעה בחולון הרכב
- ג. המערכת מנתחת את נתוני החישנים
- ד. ע"פ עוצמת המכנה בחולון המערכת מבצעת החלטה כיצד לפעול
  - ה. המערכת שולחת הודעה למרכז הבקרה (במקרה של זיהוי פריצה ממשי)

###### 1.8.1.2. **תרחיש תאונה - פגעה בגוף הרכב**

- א. הרכיב חונה - כבוי
- ב. המערכת זיהתה פגעה בגוף הרכב
- ג. המצלמה מתחילה לפעול (כאשר עוצמת הפגעה עברה את הסף להפעלה)
- ד. המערכת מנתחת את נתוני החישנים
- ה. ע"פ עוצמת המכנה ברכיב המערכת מבצעת החלטה כיצד לפעול
  - ו. המערכת שולחת הודעה למרכז הבקרה

###### 1.8.1.3. **תרחיש התדרדות של הרכיב מחנית בירידה (מצב A)**

- א. הרכיב חונה - כבוי
- ב. חייני התנועה זיהו תזוזה של הרכיב
- ג. המערכת מנתחת האם התנועה היא רגעית או רציפה
- ד. ע"פ אופי התנועה המערכת מקבלת החלטה כיצד לפעול
  - ו. המערכת שולחת הודעה למרכז הבקרה

**1.8.2. תרחישי ניהול מערכ קורנות:**

**1.8.2.1. תרחיש האטת מהירות**

- א. הקרון בנסיעה
- ב. חיישני הקרון זיהו האטה ב מהירות
- ג. הקרון מדווח למרכז הבקרה על מהירותו
- ד. ע"פ מהירות הקרון מרכז הבקרה מחליט כיצד לפעול
- ה. מרכז הבקרה שולח פקודות האצה לקרון המאט
- ו. חיישני הקרון זיהו את מהירותו העדכנית
- ז. הקרון מדווח למרכז הבקרה על מהירותו החדש
- ח. ע"פ מהירותו העדכנית של הקרון מרכז הבקרה מחליט כיצד לפעול
- ט. מרכז הבקרה שולח פקודות חירום לכלל הקורנות

**1.8.2.2. תרחיש האצת מהירות**

- א. הקרון בנסעה
- ב. חיישני הקרון זיהו האצת ב מהירות
- ג. הקרון מדווח למרכז הבקרה על מהירותו
- ד. ע"פ מהירות הקרון מרכז הבקרה מחליט כיצד לפעול
- ה. מרכז הבקרה שולח פקודות האטה לקרון המאיץ
- ו. חיישני הקרון זיהו את מהירותו העדכנית
- ז. הקרון מדווח למרכז הבקרה על מהירותו החדש
- ח. ע"פ מהירותו העדכנית של הקרון מרכז הבקרה מחליט כיצד לפעול
- ט. מרכז הבקרה שולח פקודות חירום לכלל הקורנות

**1.8.2.3. תרחיש שינוי מקום אחד הקורנות (סטייה מהמסלול/תקיעה במסלול)**

- א. הקרון בנסעה
- ב. חיישני המיקום זיהו שינוי במיקום הקרון
- ג. הקרון מדווח למרכז הבקרה על מיקומו
- ד. מרכז הבקרה זיהה חריגה ממיקומו הרצוי (עכירה/סטייה מהמסלול)
- ה. נשלחת פקודות עצירת חירום לכלל הקורנות

## 2. פרק 2 – סקר ספרות

### 2.1. מבוא לפרק

בסקירה זו נתייחס לשאלות המחקר המרכזיות הכוללות את הגישות והשיטות השונות בהנדסת מערכות.

נבחן את מידת הקשר בין ייעילות הנדסת המערכת לשיטה בה משתמש מהנדס המערכת ולשכבה בה הוא משתמש בשיטה זו.

הסקירה תהווה מתחווה עבור מה שקרה היום בעולם הנדסת המערכות, כיצד מתנהל פרויקט, באיזה שיטות משתמשים מהנדסי המערכת ומהן השיטות היעילות ביותר. מטרת העל של מחקרנו היא היכולת לאפיין לפי סוג פרויקט את השיטה בה עדיף מהנדס המערכת להשתמש ובאיזה רמה להשתמש בכל שיטה ושיטה.

כמו כן, כפי ש(Ober, Ileana & Ober, Iulian & Dragomir, Iulia & Aboussoror, El. (2011) דנו במאמרם, בשנים האחרונות חלה עלייה ניכרת בשימוש במידול בתעשייה. אחת הסיבות לכך שהשימוש במודלים מעוניין היא האפשרות לבצע אימוט, תיקוף ובדיקות מוקדמות, אולם עבודה רבה נדרשת לביצוע עבור תוצאות תיאורתיות של אימוט ותיקוף. הטמעת התוצאות האלה בתיכון מודול גדול רחב הינה ממשימה מתוגרת ביותר, שכן תוצאות אלה הן תוצאות תיאורתיות בעלות קנה מידה רחב מאוד.

בנוספ', במחקר זה נעה על השאלה מהי שיטת המידול האידיאלית עבור הנדסת מערכות ובאיזה שכבה יש לבצע – בפרויקט זה בחרנו את כל המידול והשיטות הבאות:

1. System Composer with ECSAM
2. Enterprise Architect with SysML
3. Visio with ECSAM

שאלה זו לא נבדקה במחקריהם ובמאמריהם קודמים ובמחקר זה ניתן לה תשובה על בסיס תוצאות המחקר.

כפי שהציגנו קודם לכן, מטרתנו במחקר היא לאפשר מהנדסי המערכת להגיע ליעילות מקסימלית ע"י שימוש בכלים מידול שונים.

במחקר זה נבחן כל מידול המשתמשים בשיטות שונות ואת השימוש בכלים מידול וסימולציה והשילוב ביניהם.

בכל אחת מהשיטות נבחן שתי מערכות מסוימות שונות, נדגום את הפרויקטים עפ"י קרייטריוונים ונבחן אותם.

בסוף התהליך נוכל לספק תשובה באיזה מערכות ובאיזה שיטה עדיף להשתמש עפ"י סוג המערכת.

## 2.2. שיטות וכלי מידול

הסיקור בסעיף זה מבוסס על [26],[12],[9],[35],[34],[27],[11],[25]

קיימות שיטות ושפות מידול רבות, וכיימים כל מידול רבים באמצעות ניתן לביצוע מידול ע"פ עקרונות שיטות ושפות המידול השונות.

בפרויקט זה בחרנו ב-3 כלים למידול וב-2 שיטות:

Modeling tools: System Composer, Enterprise Architect

Modeling languages: ECSAM, SysML

Diagram tools: Visio

בפרויקט שלנו הקשרים בין שיטות/שפות כל מידול וכלי השירות הם:

System Composer with ECSAM .1

Enterprise Architect with SysML .2

Visio with ECSAM .3

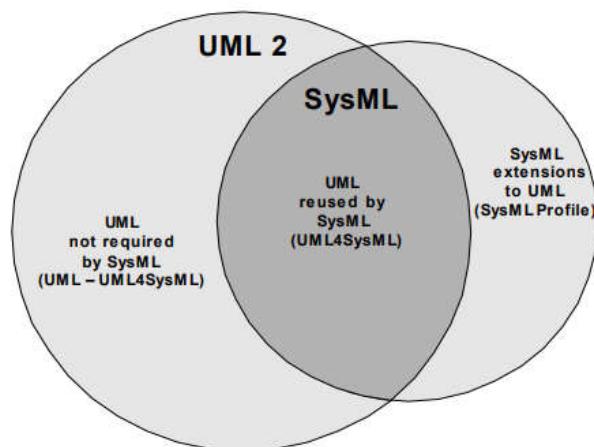
### 2.3. שפות מידול

ראשית נסקור את שתי השיטות העיקריות בהם נתמקד במחקר זה.

#### 2.3.1. שפת המידול (SysML)

SysML היא שפה המשמשת למידול מערכות. SysML תומך במפרט, ניתוח, תכנון, אימוץ ותיקוף של מגוון רחב של מערכות מורכבות. מערכות אלה עשויות לכלול חומרה, תוכנה, מידע, תהליכיים, כוח אדם ומתקנים. מ-Object Management Group, Inc. (OMG) אנו למדים כי SysML הינו התאמה של UML (Unified Modeling Language) ליישומי הנדסת מערכות. באופן דומה לדרך בה UML איחד שפות מידול המשמשות בעולם התוכנה, כך SysML נועד לאחד את שפות מידול המערכות מהנדסי מערכות. מכיוון ש-SysML משתמשת ב-UML כבסיס, מהנדסי מערכות המודלים באמצעות SysML ומהנדסי תוכנה המודלים עם UML יכולים לשתף פעולה במידול מערכות עתירות תוכנה. עובדה זו משפרת את התקשרות בין בעלי העניין השונים המשתתפים בתהליך פיתוח הממערכות ומקדמת את שיתוף הפעולה ביניהם.

באייר 2.1 ניתן לראות את הקשרים בין UML ל-SysML:



(OMG Systems Modeling Language (OMG SysML™), Version 1.3, 2012) : 2.1

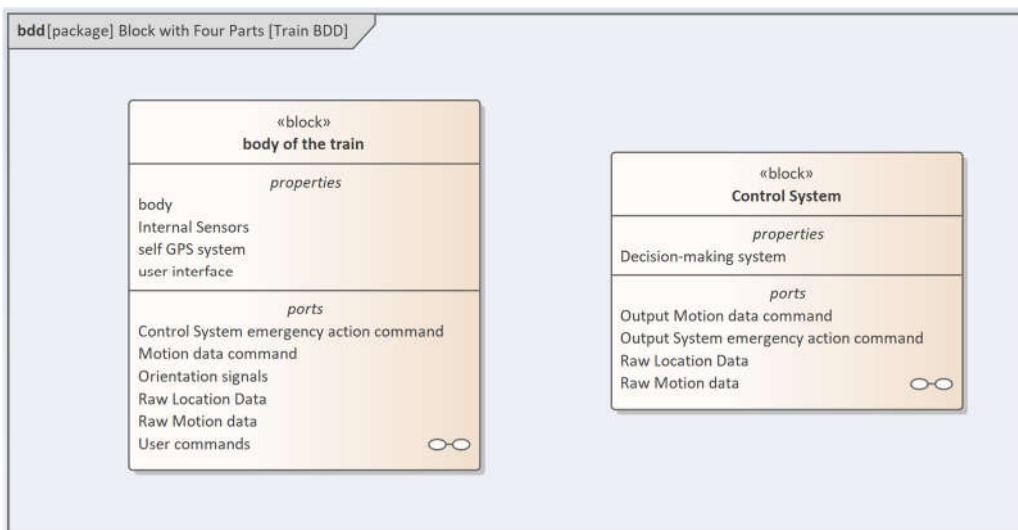
שני אלמנטים חשובים ב- SysML עליהם נפרט ובהם בוצע שימוש נרחב בפרויקט זה הם BDD ו- IBD:

### 1. Block Definition Diagram (BDD)

בלוק (block) היא יחידת מבנה מודולרית ב- SysML שמשמשת להגדרת סוגים של ישוויות פיזיות (למשל מערכת, חלק מרכיב במערכת, מערכת חיצונית או פריטים הזרמים במערכת), כמו גם ישוות רעיוניות או הפשטות לוגיות.

תרשים ה- BDD משמש להגדרת המאפיינים של כל בלוק מבחינת התכונות המבניות וההתנהגויות שלו, והיחסים בין הבלוקים וההיררכיה ביניהם.

באיור 2.2 ניתן לראות דוגמה ל-BDD של מערכת



איור 2.2: דוגמה ל-BDD מכלי מידול Enterprise architect

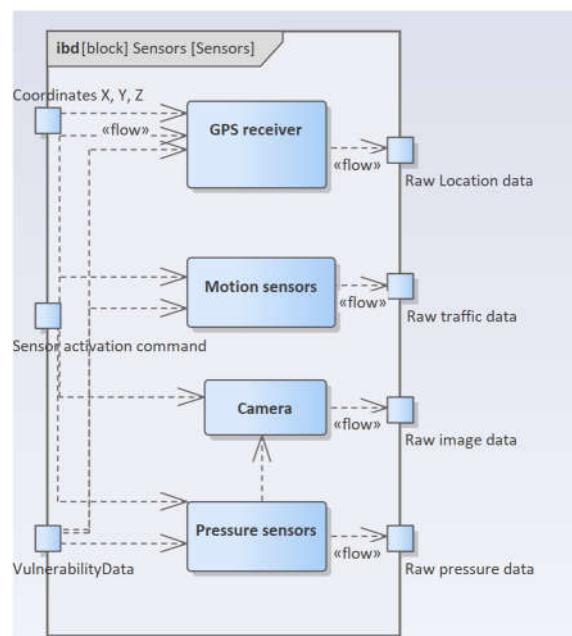
בדוגמא הנ"ל ניתן לראות שני בלוקים אשר מייצגים את תת המערכות במערך קרון חכם.

Body of the train – מייצג את הקרון על שלל תכונותיו וממשקים (O/I).  
 Control System – מייצג את מרכז הבקרה על שלל תכונותיו וממשקיו (O/I).

## Internal Block Diagram (IBD) .2

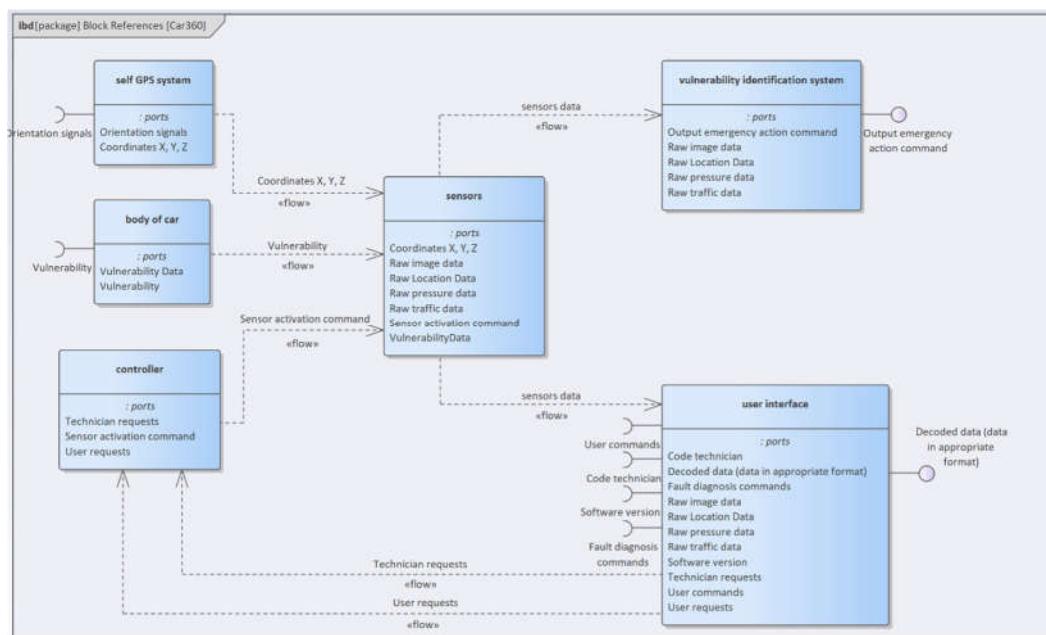
דיאגרמת ה- IBD משמשת לתיאור המבנה הפנימי של חלקו הבלוק, הצגת הקשרים בין חלקיו ומה זורם ביניהם.  
 פריטי הזרימה מייצגים ממשகים פנימיים.  
 בתרשימים מוצגות גם אינטראקציות עם בלוקים חיצוניים כדי לציין את הממשקים החיצוניים הדרושים.  
 ניתן להשתמש ב- IBD לתיאור הקשרים בבלוק יחיד (תת מערכת) ולמערכת כולה.

באיור 2.3 ניתן לראות דוגמה ל- IBD של בלוק יחיד (תת מערכת – Sensors)



איור 2.3: דוגמה ל- IBD של בלוק במערכת מכל מודול Enterprise architect

#### באיור 2.4 ניתן לראות דוגמה ל- IBD של המערכת הכוללת



איור 2.4: דוגמה ל- IBD של מערכת מכלים מידול Enterprise architect

ניתן להתייחס אל ה- BDD כאל "קופסה שחורה" אשר בה מוצגות את תכונות הבלוק, ואל ה- IBD כ"קופסה לבנה" בה מוצגים הקשרים בתוך הבלוק (או בין מספר בלוקים שונים - כאשר הבלוק המכיל הוא כלל המערכת).

דוגמה לישום ניתן לראות לפי Faida Mhenni, Jean-Yves Choley, Olivia Penas, Regis Plateaux, Moncef Hammadi (2014), שהציגו מתודולוגיה המורכבת משני שלבים:

1. ניתוח קופסה שחורה עם נקודת מבט חיצונית המספקת דרישות מקיפות ועקבית.
2. ניתוח קופסה המתאר את הארכיטקטורה ואת התנהוגות פנימית של המערכת.

ה- SysML היא שפה לשימוש כללי עבור יישומים הנדסיים למערכות. היא תומכת בModelProperty, ניתוח, תכנון, אימוץ ותיקוף של מגוון רחב של מערכות ותתי-מערכות. ב- SYSML ישנו 9 סוגי דיאגרמות, כאשר אלו התנסינו והשתמשנו בדיagramות מסווג IBD ו- BDD. כאשר BDD משמש להגדרת תוכנות מבניות והתנהוגות של יישויות פיזיות ולוגיות במערכת ו- IBD משמש לתיאור הקשרים ואופני זרימת המידע.

באמצעות דיאגרמות אלה ניתן לתאר את תכונות המערכת ואופני זרימת מידע בתוכה, כאשר ה-SYSML מסוגלת למודל מגוון רחב של מערכות, העשוויות לכלול חומרה, תוכנה, מידע, תהליכיים, כוח אדם וمتקנים.

### 2.3.2. **שיטת המידול** **Embedded computer system analysis and modeling** (ECSAM)

כפי ש-(2005) Lavi, Jonah Z. & Kudish, Joseph. הסוקרים את השיטה במאמרם ECSAM היא שיטה להצגתדרישות הנדסיות ומודלים למערכות מבוססות מחשב אשר פותחה בשנתה ה-80, בחלוקת בתעשייה האוירית.

מודל ה-ECSAM מציג מערכת באמצעות שלוש תצוגות (Module Chart, Activity Chart, StateChart) עקביות ומשלימות העוסקות בארכיטקטורה הריעונית, ביכולות המערכת, ובאופני הפעולה, שהם הבסיס הלוגי לזרינות היכולות.

באמצעות ECSAM ניתן לייצר באופן שיטתי את התרחישים התפעוליים של המערכת. טכניקת המידול מתאימה לניתוח מערכות כמו גם תת מערכות ברמה נוספת.

השיטה מדrica את המפתח בגישה דרישות המערכת, שיוכן לאלמנטים הלוגיים של המודל (יכולות, זרימות, אופני פעולה, מעברים) והקצתן לאלמנטים המבנאים של המודל (תת המערכת).

כמו כן, ECSAM משתמש בצורה פורמלית בשפת ה-StateChart של פרופ' דוד הראל (mboss על [5]) לניתוח וסימולציה.

## 4.2.4. **כלי מידול**

### System Composer .2.4.1

הסיקור בסעיף זה מבוסס על [33]

,MATLAB System Composer (SC) הוא כלי מידול של חברת Mathworks, יצרנית MATLAB וככזה הוא יכול להשתמש ביכולות והינו אחד מגוון הכלים שכללים ב- MATLAB. הרבות והמגוונות של MATLAB. לפि אתר החברה הכליל מאפשר הגדלה, ניתוח ובניה של ארכיטקטורות וקומפוזיציות להנדסת מערכות מבוססות-מודלים ותוכנו תוכנה. באמצעות כלי זה ניתן להציג דרישות תוך שיפור ארכיטקטורת המודל, את הארכיטקטורה ניתן לתכנן ולסמלץ באמצעות Simulink (כלי תכנון וסימולציה). כמו כן, ישנו כלי נוסף, State flow אשר מתחמש עם ה- SC ומאפשר תכנון ופיתוח של תזמון משימות, ניהול תקלות, פרוטוקולי תקשורת ועוד, ועובד בסימולץ המודל שבנוינו, בנוסף הוא מספק שפה גרפית הכוללת דיאגרמות מעבר, תרשימי זרימה, טבלאות וסימולציות השונות שנבחנות.

### Enterprise Architect .2.4.2

הסיקור בסעיף זה מבוסס על [6], [23]

כלי מבית Sparx Systems המשמש ליצירת מודלים חזותיים ותכנון כלים מבוסס UML. לפि אתר החברה הכליל משמש בנוסף להנדסת מערכות לשימושים רבים כמו מידול תהליכי עסקים, מידול תעשיית וכו'. יכולת הכלי הם ניתוח, תכנון, יישום, בדיקה ותחזוקה של מודלים ברמה גבוהה באמצעות שימוש ב-UML, SysML, BPMN וסטנדרטים נוספים. Enterprise Architect הוא כלי גרפי ורב-משתמשים המאפשר לצוות לעבוד ביחד על המערכת ליצירת מערכת רובוטית. בנוסף מפיק הכלי דיווחים ודוקומנטציה האחראים ניתן לעקוב ולשתף بكلות. יכולת נוספת של הכלי הוא הדמיה דינמית של המודלים, המאפשרת למהנדס המערכת הבנה טובה יותר על האופן שבו המערכת עובדת.

### Visio . 2.4.3

הסיקור בסעיף זה מובוס על [37], [1], [20]

בפועל בו אנו חיים קיימות תוכנות שרטוט רבות, חלקם תוכנות קוד פתוח מתקדמות מאוד בעלות מגוון רחב של יכולות (לדוגמה: ArgoUML, BOUML ועוד) וחלקו קוד סגור לדוגמת Visio .  
למרות היתרונות הרבים בכל השרטוט מעולם הקוד הפתוח חברות רבות מעדיפות שלא להשתמש בהן וזאת מכיוון שאין עליהם בקרה לעומת מערכות קוד סגור של חברה מוכרת שיש עליה בקרה, ותמכה ובעיקר בעיות אבטחת מידע.

عقب כך, החלפנו בפרויקט זה להתמקד בכל ה- Visio אשר נפוץ מאוד בקרב מהנדסי המערכת והחברות השונות.

ה- Visio הוא כלי שרטוט מבית Microsoft, לפי אתר החברה הכללי מאפשר ייצור תרשימי זרימה, דיאגרמות, תרשימים ארגוניים, תוכניות קומה, עיצובים הנדסיים ועוד.

### 2.5. כלים המידול בשילוב שפות המידול

במחקר זה נבדל 3 מערכות באמצעות כלים המידול שסקרנו קודם:

1. Enterprise Architect עם שפת המידול SysML

2. ECSAM לפי שיטת המידול System Composer

3. Visio עם שיטת המידול ECSAM

ע"י עבודה בשלושת הכלים ובחינת התוצאות עפ"י קרייטריונים נוכל לסקור ולהציג את היתרונות והחסרונות של עבודה בכל אחד מכל המידול עם שיטה/שפה באמצעותה מתבצע בו המידול.

## 6.2. מצב היישום בתעשייה בנושא סימולציה

הסיקור בסעיף זה מבוסס על [8], [21], [30], [31], [32], [36]

המאמצים למידול וביצוע סימולציה מערכית מתקיים כל זמן. כפי שניתן לראות בפרויקט של (Mangeruca, L. (2012, May 30) *System of System (SoS)* – מאמצים אלו התרטטו אונלייניות המבוססות על סימולציות תוך כדי הפרויקט, כגון מודלים סטטיסטיים, טכניקות אופטימיזציה, אבחונים ותחזיות. מסגרת הסימולציה כוללת תמייה בדיניות, א-סינכרוניות, חוסר עקביות ועוד. כפי שניתן לראות עפ"י (Feldman, Y. (2014) *ביצוע סימולציה באמצעות כלי מידול עפ"י שפת SysML*, הינו מאמץ מתmeshן הן בהיבטים הלוגיים והן בהיבטים הפיזיקליים. כלים מסוג זה יכולים להיות בעלי סמנטיקה התנהגותית בסיסית שונה, דבר היוצר קושי רב כאשר מבצעים שילוב בין מספר כלים במידול הפרויקט. אתגרים אלו מرتبطים בזיהוי חוסר התאימות וביצוע ההתאמות על מנת לשמר על הסמנטיקה.

כאשר אנו מגדירים בлок IBD פרטני בתוך בлок BDD כללי מוגדרת לו סמנטיקה התנהגותית (הכוללת: כנישות, יציאות, תוכנות וכו') אשר לפיהם הוא צריך לפעול. אולם כאשר ב-IBD הכללי של המערכת אנחנו משתמשים במספר IBDs פרטניים מאותו סוג ובודנים את התנהגותם בסימולציה מלאה ובשימוש פיזי, אנו מבחינים בשינוי בתנהגותו ככלומר ישנו הבדל בין אם הוא פועל באופן עצמאי לבין אם הוא פועל עם עוד IBDs (אותו סוג שלו).

כמו כן, הציפייה הייתה שנוכל להשתמש במספר IBDs זהים כשם מחוברים באותה הצורה שהוא מחוברים במודל המערכת בצורה פרטנית כשהתנהגות שלהם לא משתנה.

סיבה אפשרית היא שינוי ערך ב-port SysML flow העשי לגורור לשינוי פנימי, אירוע זה יכול להיות הטיגר לרוץ אירועים נוספים (שינויים בסימולציה כמו בעולם האמתי לא תמיד יהיה סינכרוניים ו/או יגעו אחד אחרי השני, וכן ניתן לשנות אחד בטל פעולה שינוי אחר היה צריך לבצע).

בכדי להתגבר על זה נdag שסדר השינויים לא יהיה משנה עבור הבלוקים, אפשרות נוספת היא לכפות שסדר האירועים לא ישנה.

כלי נוסף בחרכנו היה ה- SC (System Composer) של חברת Mathworks בו ביצענו מידול ע"פ שיטת ECSAM. ה- SC הוא כלי הנמצא בשימוש נרחב בתעשייה (בחברות כגון: Rolls Royce, Turkish Aerospace התעשייה ע"פ צרכים ודרישות שהועלו במהלך הפיתוח (יכולות חדשות מתווספות אופן זהה, כמו גם במחקרנו. תוך כדי המחקר העלנו מספר דרישות שהועברו לפיתוח וחלקים אף יופיעו כבר

בגרסאות הקרובות – לדוגמה נושא הדוחות, שרטוטים ותרשיים).

כפי ש Matt Ley (2019) הציג יישום תהליכי פיתוח פרויקט באמצעות שילוב הממערכות SC ו- Simulink אשר אפשרה לROLS רvais לבצע חשיבה חדשה היכן צרך להוסף פירוט והיכן ניתן להימנע מפירוט יתר.

בנוסף , Shipton, L. (2017). מציג את ההשוואה בין העבודה של מהנדס המערכת עם כל ה- SC לעובדה ב淵דיון, בפרויקטים מסוג (SoS) System of System

שלבי עבודה הנדרשת המערכת ללא הכלים: ריכוז רשות דרישות, שרטוט הארכיטקטורה (קשה עד בלתי אפשרי לעקוב אחריו כל הדרישות בבנייתה), ביצוע מידול לארכיטקטורה בהתאם לדרישות,

קביעת ארכיטקטורה אופטימלית/מקובלת, ובמקביל ביצוע אנליהזה ל- SoS, תכנון דגמי המערכת בהתאם לדרישות, עיצוב דגמים שמקורם בדגמי הארכיטקטורה, ביצוע בדיקות שדה מלאות לכל ה- SoS ולמעשה רק בשלב זה אנו מבצעים בדיקות מקיפות לכל ה- SoS.

באמצעות סט הכלים של Mathworks אנו מקבלים יכולות נוספות אשר עוזרות לנו בהנדסת המערכת.

כל ה- SC מאפשר לנו יבוא/בנייה של דגמי הארכיטקטורה, הגדרת מאפייני הארכיטקטורה עם Profiles ו-Stereotypes , ניתוח הארכיטקטורה, חיבור ישיר בין מידול הארכיטקטורה ועיצוב הדגמים.

כל ה- Simulink מאפשר לנו כתיבה/יבוא של דרישות, יכולת מעקב דו-כיוונית בין מידול הארכיטקטורה, עיצוב הדגמים והבדיקות.

שילוב הכלים בתוספת כל ה- State flow מאפשר לנו תכנון מערכות מרובות תחומיים (חומרה, תוכנה וכו' ), ביצוע אימונות מוקדם ע"י ביצוע סימולציה וניתוחים סטטיסטיים, ובכך למעשה מאפשר לנו לעבוד בצורה מסודרת יותר (בכל אחד אשר מכיל את כל המידע תוך סינכרון וקורלציה בין הדרישות והתכנון) ויוטר מכל לבצע סימולציות וניתוח של המידול והעיצוב בשלב מוקדם משמעותית.

כמו כן, ישנה חשיבות רבה בתעשייה לשילוב של הנדסת מערכות ותהליכי בקרה, כאשר ארכיטקטורת מערכות הבקרה נקראת SCADA (Supervisory Control And Data

. SCADA הן מערכות מחשב המשמשות לצורך פיקוח, שליטה ואיסוף נתונים.

מערכות ה- SCADA מנטרות ושולטות על תהליכי בתחום התעשייה, תשתיות, ניהול מתקנים ועוד.

מערכת ה- SCADA מורכבת בדרך כלל ממתת המערכות הבאות:

1. ממשק משתמש - באמצעותו המפעיל/משתמש מנטר ושולט על התהליך.
2. מערכת פיקוח ממוחשבת - אוסףת את המידע ושולחת פקודות בקרה.
3. יחידות ניטור בתהיליך (RTU) - יחידות קצה ממוחשבות המחויבורות לחישנים.
4. בקרים לוגיים מיתכנתים (PLC) - מכשירים אלקטרוניים לאיסוף מידע.
5. תשתיית תקשורת המחברת בין מערכת הבקרה ליחידות הקצה.

אחת הדוגמאות בהן ניתן לראות את מערכת ה- SCADA בפרויקט זה היא בדוגמה "ניהול מערכ קرونות" כאשר:

- ממשק המשתמש מקביל ל- User Interface המשמש לניטור ושליטה ידנית על התהליך.
- מערכת פיקוח ממוחשבת מקבילה ל- Decision-making system אשר אוסףת את נתוני הקرونות ושולחת פקודות בהתאם למתרחש.
- יחידות ניטור בתהיליך (RTU) מקבילות ל- Internal Sensors בקרונות אשר אוספות מידע ומעבירות למערכת הפיקוח הממוחשבת.

בנוסף, כפי שנאמר ע"י Transentis Consulting, בעולם הפיקת הדוחות ישנו ניסיון לשנות את דפוסי העבודה של המהנדסים מכתיבת דוחות ידנית למעבר להפקת דוחות מתוך כלי המידול, השתחררות מ- Document-driven (כתיבה מסמכים ידנית) ל- Process-oriented (הפקת דוחות מתוך כלי המידול).

## 7. סקירות החישונים

### 2.7.1. חישון לחץ

הסיקור בסעיף זה מבוסס על [3],[13],[28]

בעבודה זו חישון הלחץ ישמש אותנו עבור הדוגמה אשר עוסק ב"ניטור פגיעה ברכב 360" וזאת על מנת לתאר מכה בעוצמות שונות בגוף הרכב, החישון יקבל את נתוני הפגיעה ויחזיר ערכיים של 0, 1 ו-2 על פי עוצמות הפגיעה הנמדדות, ערכיים אלה ישלחו אל המערכת אשר תשלլ את נתוני כל החישונים בהתאם לכך תבצע את הפעולה המתאימה.

תיאור חישון לחץ מפורט בנספח א-1

### 2.7.2. חישון מהירות

הסיקור בסעיף זה מבוסס על [2]

בעבודה זו חישון המהירות ישמש אותנו עבור הדוגמה אשר עוסק ב"ניטור פגעה ברכב 360" ובדוגמה אשר עוסקת ב"ניהול מערכ קرونוט חכם" וחאת על מנת לתאר את מהירות הרכיב/ הקרון

בדוגמה העוסקת ב"ניטור פגעה ברכב 360", החישון ישמש למדידת מהירות הרכיב (עקב גיבתו/התדרדרות) החישון יקבל את נתוני המהירות ויחזיר ערכיים של 0, 1 ו-2 על פי המהירות הנמדדת, ערכיים אלה ישלחו אל המערכת אשר תשלל את נתוני כל החישונים בהתאם לכך תבצע את הפעולה המתאימה.

בדוגמה העוסקת ב"ניהול מערכ קرونוט חכם", החישון ישמש למדידת מהירות הקرونוט (הגברת/האטת מהירות)

החישון ימדד וישלח את נתוני המהירות הנמדדת אל המערכת אשר תשלל את נתוני כל החישונים עבור כל הקرونוטים ובהתאם לכך תבצע את הפעולה המתאימה.

תיאור חישון מהירות מפורטBNספח א-2

### 2.7.3. מצלמות 360

הסיקור בסעיף זה מבוסס על [4], [15]

בuebaודה זו המצלמה תשמש אותנו עבור הדוגמה אשר עוסק ב"ניסיור פגיעה ברכב 360" וזאת על מנת לצלם את האירוע/ הסביבה בזמן אמת בעת מכיה בעומת חזקה (ישנם קритריונים מה היא מכיה חזקה - סעיף 3.2.3 – "טבלת תרחישים והסביר מדוע חישנים:"), החישון מקבל את הנתון יישירות מהישן הלחץ בטוחה ערכים של 2-0 על פי עוצמות הפגיעה הנמדדות (ערכים אלה יקבעו האם להפעיל את המצלמה או לא) התמונות המצלומות ישלו אל המערכת וממנה אל מרכז הבקרה.

תיאור מפורט נמצא בנספח א-3

### 2.7.4. מקלט GPS

הסיקור בסעיף זה מבוסס על [12], [16], [29]

בuebaודה זו מקלט ה-GPS ישמש אותנו עבור הדוגמה אשר עוסק ב"ניסיור פגעה ברכב 360" ובדוגמאות אשר עוסקת ב"ניהול מערכ קرونוט חכם" וזאת על מנת לתאר את מיקום הרכב/ הקרון

בדוגמה העוסקת ב"ניסיור פגעה ברכב 360", החישון ישמש למדידת המיקום בזמן אמת ויקבע האם חל שינוי במיקום, כלומר אם הרכב זו ממקומו (עקב התדרדרות/ גנבה) החישון יקבל את נתוני המיקום ויחזר ערכים של 0, ו-1, כלומר זו ממקומו או לא, ערכים אלה ישלו אל המערכת אשר תשקל את נתוני כל החישונים בהתאם לכך תבצע את הפעולה המתאימה.

בדוגמה העוסקת ב"ניהול מערכ קرونוט חכם", החישון ישמש למדידת המיקום בזמן אמת ויקבע האם חל שינוי במיקום הקرونוט שלא עפ"י התנאים (סטיטה מהנתיב/ תקיעת קרון) החישון ישלח את נתוני המיקום אל המערכת אשר תשקל את נתוני כל החישונים עבור כל הקرونוטים ובהתאם לכך תבצע את הפעולה המתאימה.

תיאור מקלט GPS מפורט בנספח א-4

## 8. סיכום הפרק

בסקירה זו התייחסנו לכל שאלות המחקר ולמטרת המחקר, כפי שניתן לראות עפ"י. סקירה זו אנו מציגים את הנתונים הידועים היום בעולם הנדסת המערכות. הנתונים הידועים היום אינם נתונים תמונה מלאה ואינם יכולים להוביל ליעילות מקסימלית בתהליכי של הנדסת המערכת. במחקר זה ננסה להשלים את הפרסומים הנוכחיים ולספק מידע אשר באמצעותו ניתן יהיה ליעיל את הנדסת המערכות בפרויקטטים עתידיים ולהתקרב ליעילות מקסימלית. כמו כן, ננסה להראות את החשיבות שבמידול כבר בשלבים הראשונים והיתרונות אותם מהנדסי המערכת יכולים להפיק מימוש שיטות אלו.

### 3. פרק 3 – מתודולוגיה

#### 3.1. הפרדיגמה המחברת

רצינול וענין עובdot מחקר זו, כולם שאלות המחקר המרכזיות, הם ההשווואה בין מספר שפונות/שיטות מידול בכליים שונים וההשלטה מהו השימוש הייעיל ביותר עבור מהנדס המערכת וזאת על פי מדרדים לבחינת שאלת המחקר לצרכי הנדסת מערכות. כמו כן, נבחן עד איזו רמה כדאי להעמיק במבנה המודלים וביצוע הסימולציות בפרויקט.

המחקר כולל שלושה חלקים עיקריים:

1. בוחנת שיטות וכליים על בסיס המידע בספרות הטכנית
2. לימוד עמוק של כלים ומידול מערכות וסימולציה באמצעות הכלים והשיטות שבחרנו למחקר השוואתי
3. הצגת תוצאות המדרדים שנבחרו למען לשאלות המחקר, על בסיס הפעולות המعيشית של סעיף 2 לעיל

מרחיב השיטות והכלים הקיימים רב מאוד. אי לכך נדרש סינון וראשוני של אותן השיטות והכלים וזאת על מנת לשמור על היקף מציאותי של עבודה המחקר, ולא פחות חשוב מזה להביא תוצאות שימושיות למתנדסי המערכות על מנת שיוכלו להשתמש בעבודה זו ובפרסומים הצפויים בספרות כתוצאה ממנה.

אי לכך נבחרו כלים ושיטות אשר נמצאים בשימוש במספר לא מבוטל בארגוני פיתוח בארץ:

Modeling tools: System Composer and Enterprise Architect

Modeling languages: ECSAM and SysML

Diagram tools: Visio, and MS Office

בפרויקט שלנו הקשרים בין שיטות/שפונות כלים המידול וכלי השירות הם:

System Composer with ECSAM .1

Enterprise Architect with SysML .2

Visio with ECSAM .3

כמו כן, כפי שצוין בפרק 1, פרק המבוא למחקר, שאלות המחקר הראשיות והמשניות הן:

#### שאלות מחקר ראשית

1. מהי שיטת המידול האידיאלית עבור הנדסת מערכות ובאיזה שכבה יש לבצע?
2. מה יעיל יותר עבור מהנדס המערכת, כדי מידול מלאה בשפה/שיטת ידועה בעל יכולת קישור אל הפיזיקה של חלקים המערכת בתוך אותו כלי או שילוב של כלים מידול וشرطוטים מערכתיים שונים?

#### שאלות מחקר משנהות

3. באיזה מידה יש יתרון למידול כלים על פני שימוש בשרטוטי Visio?
4. כאשר מבצעים סימולציה, באיזה שכבה מערכת יש לבצע את הסימולציה (באיזה רמה יש לבצע את התறחישים)
5. עד איזה שכבה יש ערך נוסף לביצוע הסימולציה?

**להלן טבלה 3.1 המציגת קורלציה בין שאלות המחקר לבין המדרדים שנמדדו בפרויקט:**

מדד	שאלה מחקר	מדד מס'
1. נוחיות ויזואלית 2. שפת מידול מוכרת לכל המשתתפים בפרויקט 3. ביצוע הגדרת חוזרת / עדכון הגדרה 4. OUT\IN - שינוי ניסות ויציאות מהמערכות השונות 5. שינוי ארכיטקטורה באמצעות הוצאות אלמנטים 6. השפעה של שינויים באמצעות הפרויקט	מהו שיטת המידול האידיאלית עבור הנדסת מערכות ובאיזה שכבה יש לבצע?	1
1. נוחיות ויזואלית 2. ביצוע הגדרת חוזרת / עדכון הגדרה 3. OUT\IN - שינוי ניסות ויציאות מהמערכות השונות 4. שינוי ארכיטקטורה באמצעות הוצאות אלמנטים 5. השפעה של שינויים באמצעות הפרויקט	מה ייעיל יותר עבור מהנדס המערכת, כדי מידול מלאה בשפה/שיטת ידועה בעל יכולת קישור אל הפיזיקה של חלקי המערכת בתוך אותו כלי או שילוב של כלים מידול וشرطוטים מערכתיים שונים?	2
1. נוחיות ויזואלית 2. ביצוע הגדרת חוזרת / עדכון הגדרה 3. OUT\IN - שינוי ניסות ויציאות מהמערכות השונות 4. שינוי ארכיטקטורה באמצעות הוצאות אלמנטים 5. השפעה של שינויים באמצעות הפרויקט	באיזה מידה יש יתרון למידול כלים על פני שימוש בשרטוטי Visio?	3

**טבלה 3.1: קורלציה בין שאלות המחקר לממדים**

להלן טבלה 3.2 המיצגת קורלציה בין המדרדים שנמדדו בפרויקט לבין דרך המדידה שלהם:

Enterprise Architect	System Composer	Visio	דרך מדידה	מדד	מספר מדד
V	V	V	הציג ארכיטקטורת המערכת כולל תת המערכות השונות בה	נוחיות ויזואלית	1
V	V	V	הגדרה חדשה (באופן שונה) של אלמנט מרכזי וSCPOL תת מערכות	ביצוע הגדרת חזרה / עדכון הגדרה	2
V	V	V	מחיקה, הזזה ושינויי של כניסה ויציאות מהמערכות השונות	IN\OUT - שינוי כניסה ויציאות מהמערכות השונות	3
V	V	V	הזזה (שינוי מיקום) של אלמנטים קיימים וקיים רום חדש למבנה	שינוי ארכיטקטורה באמצעות הזזה אלמנטים	4
V	V	V	בדיקות רמת מורכבות שפת המידול והקליה הנבחן כולל מושגים בסיסיים	שפה מידול מוכרת לכלל משתמשים בפרויקט	5
V	V	V	ביצוע שינויים משמעותיים בארכיטקטורה בשלב ראשוני ומתקדם	השפעה של שינויים באמצעות הפרויקט	6

טבלה 3.2: קורלציה בין המדרדים לדרכי מדידתם

### על מנת לבחון את שאלות המחקר בחרנו 2 דוגמאות שונות:

נבחרו 2 דוגמאות שמאפייניות את בעיות ה"קיצון" שמאתגרות כבר מספר שנים את עולם המידול.

#### 1. "NEYTOR פגיעה ברכיב 360" - מערכת משולבת עם סיבוכיות גבוהה בעלת תתי מערכות עם סיבוכיות ביןנית/גובה

בדוגמה זו נבחן רכיב אחד מאוד מורכב, אשר מורכב ממספר חישנים שונים. הרכיב אשר נבחן הוא גוף הרכיב אשר אליו מחוברים חישנים שונים המדווחים אל מערכת הזיהוי פגיעיות המדווחת אל מרכז הבקרה. המחקר בעצם בוחן את הכלים השונים ברמת הקושי לבצע מידול של מערכת אחת מורכבת בעלת תתי מערכות רבות.

דבר זה מסבך מאוד את בניית הסימולציה עצמה אשר מצריכה תיאור מלאה של התנהוגיות המערכת, ירידא לפרטים, בנייה של כל תשתית המערכת וההתנהוגיות השונות שלה וביצוע הסימולציה על כל תשתית מערכת בנפרד ועל המערכת כולה בשלמותה.

#### 2. "ניהול מערכת קرونות חכם" - מערכת משולבת עם סיבוכיות ביןנית/גובה בעלת הרבה תתי

##### מערכות זהות עם סיבוכיות נמוכה

בתרחיש זה נבחן מספר רב של רכיבים פשוטים המדוחים נתונים אל מרכז הבקרה אשר מהוות את מרכז העיבוד והחישוב, למעשה הוא זה שמקבל את החלטות. הרכיבים אשר נבחנו הם קرونות תחבורה אוטונומיים המדוחים נתונים אל מרכז בקרה חכם אשר מעבד את הנתונים מכל הקرونות ומבצע את ההחלטה.

- מערכת זה פועל כמו ארכיטקטורת SCADA כאשר יחידות ניטור RTU על גבי הקرونות מדוחות למרכז בקרה חכם אשר מקבל את החלטה בהתאם למתרחש ושולח פקודות בהתאם לקרונות.

כמו כן, המחקר בעצם בוחן את הכלים השונים ברמת הקושי לבצע שכפול רכיבים והתמודדות עם כלל המידע המגיע מכל רכיב בנפרד לאחר השכפול, יצירה של רכיב אחד פשוט והכפלתו ב- $n$  פעמים.

דבר זה משפיע מאוד על בניית הסימולציה עצמה אשר מצריכה תיאור מלאה של התנהוגיות המערכת, ירידא לפרטים, בנייה של תשתית המערכת והכפלתה ב- $n$  פעמים וביצוע הסימולציה על המערכת כולה בשלמותה.

- במקרה שלנו, תחילת בניית המערכת לפי קרן אחד וכשרצינו להכפיל אותו גילנו שיש שינוי גדול בין קרן 1 לבין 2 קرونות ומעלה, דבר אשר השפיע וגרם לכך שארכיטקטורת המערכת ברמה הלוגית השתנתה.

### 3.1. מבחן המידול והסימולציה

בפרק זה יוצגו התרחישים והמופעים עבור כל אחד מהדוגמאות שבייצעו עלייהם את המבחן ועליהם ביססנו את תוצאות ומסקנות המבחן

- הרטוטים המפורטים לתרחישי ה- SD עצם עבור כל דוגמה וכל מופע נמצאים בספחums עפ"י הסעיף המתאים להם (נספח ב-1 ו- נספח ב-2)
- סה"כ מבחן של 8 מופעי תרחישים פרטניים ששימשו לכל ניסוי

#### 3.1.1. ניטור פגיעה ברכיב 360

בטבלה 3.3 להלן מ羅צים התרחישים ופירוט המופעים עבור כל תרחיש שנבדקו בניסוי "ניטור פגעה ברכיב 360".

- כל ה- SD של הדוגמה הנ"ל נמצאים בספח א-1
- עץ הצלבים עבור הדוגמה הנ"ל נמצאים בסעיף 3.3

תרחיש	שם המופע	מוףע
	מופע 1	היישני הלוח זיהו מכہ בחלון הרכיב בעוצמה של 30000psi, הפגיעה לא גרמה נזק לחalon, היישני הלוח זיהו את המכہ שהייתה מתחת לסף הנמוך. עקב הפגיעה החלה היישני המצלמות לא הופלו ולא נשלהה הטראה למרכז הבקרה.
ניסוי פריצה על ידי פגעה בחלון הרכיב	מופע 2	היישני הלוח זיהו מכہ בחלון הרכיב בעוצמה של 40000psi הפגיעה גרמה לשדק בחלון הרכיב, היישני הלוח זיהו את המכہ שהייתה מעל הסף הנמוך, אך מתחת לסף הגובה, עקב הפגיעה הבינונית היישני המצלמות לא הופלו ונשלהה הטראה למרכז הבקרה.
	מופע 3	היישני הלוח זיהו מכہ בחלון הרכיב בעוצמה של 50000psi הפגיעה גרמה לשבר בחלון הרכיב, היישני הלוח זיהו את המכہ שהייתה מעל לסף הגובה, עקב הפגיעה החזקה היישני המצלמות הופלו ונשלהה הטראה למרכז הבקרה.

<p>היישני הלחץ יהיה מכח הגוף הרכיב בעוצמה של 30000psi, היישני הלחץ יהיה את המכחה שהייתה מתחת ל██ הנמוך. עקב הפגיעה החלשה היישני המצלמות לא הופלו ולא נשלחה הטראה למרכז הבקרה.</p>	<b>מופע 1</b>	
<p>היישני הלחץ יהיה מכח הגוף הרכיב בעוצמה של 40000psi, היישני הלחץ יהיה את המכחה שהייתה מעל הסף הנמוך, אך מתחת ל██ הגובה, עקב הפגיעה הבינונית היישני המצלמות לא הופלו ונשלחה הטראה למרכז הבקרה.</p>	<b>מופע 2</b>	<b>תאונה - פגיעה בגוף הרכיב</b>
<p>היישני הלחץ יהיה מכח הגוף הרכיב בעוצמה של 130000psi, היישני הלחץ יהיה את המכחה שהייתה מעל ל██ הגובה, עקב הפגיעה החזקה היישני המצלמות הופלו ונשלחה הטראה למרכז הבקרה.</p>	<b>מופע 3</b>	
<p>היישני התנועה יהיה שינוי של 1km/h ב מהירות הרכיב, שינוי התנועה היה רגעי וכותזאה מכך הרכיב זו ממקומו, היישני התנועה יהיה את שינוי המהירות שהיא מתחת ל██ הטראה, עקב השינוי הנמוך לא נשלחה הטראה למרכז הבקרה.</p>	<b>מופע 1</b>	<b>התדרדרות של הרכיב מחניתה בירידה (מצב N)</b>
<p>היישני התנועה יהיה שינוי של 5km/h ב מהירות הרכיב, שינוי התנועה היה רציף וכותזאה מכך הרכיב זו ממקומו, היישני התנועה יהיה את שינוי המהירות שהיא מעל ל██ הטראה, עקב השינוי נשלחה הטראה למרכז הבקרה.</p>	<b>מופע 2</b>	

טבלה 3.3: ניטור פגעה ברכב 360 - טבלת תרחישים ומופעים

### 3.1.2 ניהול מערכ קرونות חכם

בטבלה 3.4 להלן מ羅צים התרחישים ופירוט המופעים עבור כל תרחיש שנבדקו בניסוי ""**גיטור מערכ קرونות חכם**"".

- כל ה- SD של הדוגמה הנ"ל נמצאים בנספה א-2
- עבור הדוגמה הנ"ל לא קיים עץ צרכים וזאת מכיוון שמדובר על קרונ אחד פשוט אשר מוכפל מספר רב של פעמים لكن לא היה צורך להגדיר עץ צרכים מפורט

תרחיש	שם המופיע	מספר מופע	תיאור
	מופע 1	1	חישני התנועה זיהו האטה ב מהירות הקרון, חישני התנועה זיהו את שינוי המהירות שהייתה מתחת ל סף ההתראה, עקב השינוי הנמוך מס' ההתראה לא נשלחה פקודת האטה לקרון.
האטת מהירות	מופע 2	2	חישני התנועה זיהו האטה ב מהירות הקרון, חישני התנועה זיהו את שינוי המהירות שהייתה מעל ל סף ההתראה אך מתחת ל סף עצירת החירום של המערכת, עקב השינוי נשלחה פקודת פקודת האטה לקרון.
	מופע 3	3	חישני התנועה זיהו האטה ב מהירות הקרון, חישני התנועה זיהו את שינוי המהירות שהייתה מעל ל סף ההתראה ו מעל סף עצירת החירום של המערכת, עקב השינוי נשלחה פקודת האטה לקרון ופקודת חירום לשאר הקرونות.
	מופע 1	1	חישני התנועה זיהו האטה ב מהירות الكرון, חישני התנועה זיהו את שינוי המהירות שהייתה מתחת ל סף ההתראה, עקב השינוי הנמוך מס' ההתראה לא נשלחה פקודת האטה לקרון.
האצת מהירות	מופע 2	2	חישני התנועה זיהו האטה ב מהירות الكرון, חישני התנועה זיהו את שינוי המהירות שהייתה מעל ל סף ההתראה אך מתחת ל סף עצירת החירום של המערכת, עקב השינוי נשלחה פקודת פקודת האטה לקרון.
	מופע 3	3	חישני התנועה זיהו האטה ב מהירות الكرון, חישני התנועה זיהו את שינוי המהירות שהייתה מעל ל סף ההתראה ו מעל סף עצירת החירום של המערכת, עקב השינוי נשלחה פקודת האטה לקרון.

<p>ומעל סך עציירת החירות של המערך, עקב השינוי נשלחה פקודת האטה לcron ופקודת חירום לשאר הcronות.</p>	<div style="border: 1px solid black; padding: 5px;"></div>
<p>חישני המיקום יהיה שינוי במיקום הקرون מהמסלול המוגדר לו, חישני המיקום יהיה את השינוי ועקב כך נשלחה פקודת עציירת חירום לשאר הcronות.</p>	<b>1</b> <b>מופע</b>
<p>חישני המיקום יהיה חוסר שינוי במיקום הקرون במסלול המוגדר לו, ככלומר תקיעת הקرون על המסלול, עקב כך נשלחה פקודת עציירת חירום לשאר הcronות.</p>	<b>2</b> <b>מופע</b>

**טבלה 3.4: ניהול מערך קדוננות חכם - טבלת תרחישים ומופעים**

**3.1.3. טבלת תרחישים והסביר מדי חישנים**  
**3.1.3.1. טבלת תרחישים – ניטור פגעה ברכב 360**

בטבלה 3.5 להלן ניתן לראות את פירוט התרחישים והמופעים עם נתוני החישנים (נתונים אמיתיים) ואת תגובת המערכת עבור אותו מופע בכל תרחיש.

נתונים אלה הם "לב הפיזיקלי" של המידול המפורט והסימולציות שבוצעו בדוגמה הראשונה ("ניטור פגעה ברכב 360"), אשר מאפשרים הגיעו למקומות ולענות על שאלות המחקר, תוך התייחסות לכל המדריכים שהוגדרו.

• בסעיף 3.2.3.2 מתוארים הערכים שזורמים במודל בעקבות ערכים אלה.

פעולות המערכת	מערך חישנים					מופיע	מספר מופיע	תרחיש
	GPS	מהירות	מצולמות	לחץ				
לא נשלחה התראת למרכז הבקרה.	0	0km/h	0	30000psi	עוצמת המכחה בחalon: הרכב: psi30000 Expected :Results לא נגרם נזק לחalon	מופע	1	ניסוי פריצה על ידי פגעה בחalon הרכב
נשלחה התראת למרכז הבקרה על פגעה בעוצמה ביןונית בחalon.	0	0km/h	0	40000psi	עוצמת המכחה בחalon: הרכב: psi40000 Expected :Results סדק לחalon	מופע	2	
נשלחה התראת	0	0km/h	1	50000psi	עוצמת המכחה	מופע	3	

لمركز הbkраה על פגיעה בעוצמה גבואה בחלון.					בחלון הרכב: psi50000 Expected :Results שבר בחלון		
דיווח למרכז הbkраה ולמשתמש על פגיעה בגוף הרכב.	0	0km/h	0	30000psi	עוצמת הפגיעה: בגוף הרכב: psi30000 Expected :Results לא נגרם נזק לגוף הרכב	מופע 1	
דיווח למרכז הbkраה ולמשתמש על פגיעה חמורה בגוף הרכב.	0	0km/h	0	40000psi	עוצמת הפגיעה: בגוף הרכב: psi40000 Expected :Results נזק קל לגוף הרכב	מופע 2	תאונה - פגיעה בגוף הרכב
דיווח למרכז הbkраה ולמשתמש על פגיעה חמורה בגוף הרכב העשהיה	0	0km/h	1	130000psi	עוצמת הפגיעה: בגוף הרכב: psi130000 Expected :Results נזק חמור לגוף הרכב	מופע 3	

לכלול נפגעים.							
לא נשלחה התראה למרכז הבקרה.	0	1- 5km/h	0	opsi	הרכבת התדרדר ב מהירות: km/h1 Expected :Results הרכב נעצר לאחר מספר שניות	מופע 1	התדרדרות של הרכב מחניתה בירידת (מצב N)
נשלחה התראה למרכז הבקרה על קר שהרכב בתנועה.	0	>5km/h	0	opsi	הרכבת התדרדר ב מהירות: km/h5 Expected :Results הרכב ממשיך להתרדר עד שנעצר ע"י עצם כלשהו בדרך.	מופע 2	

טבלה 3.5: ניטור פגיעה ברכב 360 - טבלת תרחישים ומופעים עם נתוניים אמיתיים

### 3.1.3.2. טבלת ערכי החישנים בתוחשיים ניטור פגיעה ברכב 360

בטבלה 3.6 להלן ניתן לראות את כל החישונים הנמצאים במערכת "ניטור פגעה ברכב 360".  
 עבור כל חישון בטבלה מוצג הערך אשר יכול להישלח במערכת, סוף השינוי שלו (כלומר מתי משתנה הערך שלו), המצב שלו והפעולה אשר מבוצעת (שליחת התראות אל מרכז הבקרה)  
 בהתאם לפירוט הערכים הפיזיקליים בטבלה 3.5.

חישון	ערך נשלח	סוף שינוי	מצב	פעולה
GPS		דריכת המערכת	געל מיקום XYZ ע"י המערכת	
	0	old xyz = new xyz	אין שינוי במיקום הרכיב	לא נשלחת התראה
	1	old xyz != new xyz	מיקום הרכיב השתנה	נשלחת התראה על שינוי מיקום הרכיב
	0	דריכת המערכת	מתחלות לצלם ע"פ חישון החץ	
מצלמות	0	0	המצלמות לא מצלמות	לא נשלחת התראה
	1	1	המצלמות לא מצלמות	נשלחת התראה למע' זיהוי פריצות ופגיעות
לחץ		דריכת המערכת	שמירת לחץ נוכחת כ-0.05psi	
	0	0-30000psi	שינוי מינורי בלוח	לא נשלחת התראה

נשלחת התראה על שינוי לחץ ברמה ביןונית.	שינוי לחץ ברמה ביןונית	30001-60000psi	1	
נשלחת התראה על שינוי לחץ ברמה גובהה ונשלחת פקודת הפעלה למכצלמות.	שינוי לחץ ברמה גובהה	60001-15000psi	2	
	- $V=0 \text{ km/h}$ מהירות רצiosa	דריכת המערכת		
מצב תקין	מכוניות במצב רצוי	$V < 5 \text{ km/h}$	0	
התראה על רכב בתנועה	מהירות גובהה מהרצוי	$V \geq 5 \text{ km/h}$	1	מהירות

טבלה 3.6: ניטור פגיעה בדרכב 360 - טבלת ערכי חיששנים

### 3.1.3.3. טבלת תרחישים – מילוי מערך קדוננות חכם

בטבלה 3.7 להלן ניתן לראות את פירוט התרחישים והמופעים עם נתוני החישנים (נתונים אמיתיים) ואת תגובת המערכת עבור אותו מופע בכל תרחיש.

נתונים אלה הם "לב הפיזיקלי" של המידול המפורט והסימולציות שבוצעו בדוגמה הראשונה ("ניטור פגעה ברכב 360"), אשר מאפשרים הגיעו למקומות ולענות על שאלות המחקר, תוך התייחסות לכל המדדים שהוגדרו.

- בסעיף 3.2.3.4 מתוארים הערכיים שזורמים במודל בעקבות ערכיים אלה.

פעולות המערכת	מערך חישנים		מופע	מספר מופע	תרחיש
	GPS	מהירות			
לא נשלחת כל פקודה ל夸רונות.	new xyz	98km/h	האטה של km/h2 ב מהירות אחד הקرونוט. Expected Results: לא תשלח כל פקודה, שינוי מהירות נמוך מס' ההตราה.	1	מופע
נשלחת פקודת האצה ל夸ון המאט בלבד.	new xyz	90km/h	"האטה של km/h10" ב מהירות אחד הקرونוט. Expected Results: תשליח פקודת האצה ל夸ון המאט, שינוי מהירות גבוהה מס' ההตราה אך נמוך מס' החירום."		האטת מהירות
נשלחת פקודת האצה ל夸ון המאט ופקודת חירום לשאר הקرونוט.	new xyz	80km/h	"האטה של km/h20" ב מהירות אחד הקرونוט. Expected Results: תשליח פקודת האצה ל夸ון המאט ופקודת חירום לשאר הקرونוט,		

			שינוי מהירות גובה מס' התראאה ומספר החירום. "			
לא נשלחת כל פקודה לקרונוט.	new xyz	102km/h	"הגברת מהירות של km/h2 ב מהירות אחד ה الكرונות. Expected Results תשלח כל פקודה, שינוי מהירות נמוך מס' התראאה. "	מופע 2		
נשלחת פקודת האטה לכיוון המאיצ' בלבד.	new xyz	110km/h	"הגברת מהירות של km/h10 ב מהירות אחד ה الكرונות. Expected Results תשלח פקודת האטה ל קרן שմוגבר את המהירות, שינוי מהירות גובה מס' התראאה אך נמוך מס' החירום. "			הגברת מהירות
נשלחת פקודת האטה לכיוון המאיצ' ופקודת חירום לשאר ה الكرונות.	new xyz	120km/h	"הגברת מהירות של km/h20 ב מהירות אחד ה الكرונות. Expected Results תשלח פקודת האטה ל קרן שטוגבר את המהירות, ופקודת חירום לשאר ה الكرונות, שינוי מהירות גובה מס' התראאה ומספר החירום. "			
נשלחת פקודת עצירת חירום לכל ה الكرונות.	new xyz	100km/h	"שינוי מיקום לא ע"פ המסלול המוגדר.	מופע 1		שינוי מיקום אחד ה الكرונות (סתימה/תקיעה)

			:Expected Results תשלה פקודת עצירה לכל הקرونות."	
נשלחת פקודת עצירת חירום לכל הקرونות.	old xyz	0km/h	"חוסר שינוי מיקום (עצירה) של הקרון. :Expected Results תשלה פקודת עצירה לכל הקرونות."	מופע 2

טבלה 3.7: ניהול מערכ קרוונת חכם - טבלת תרחישים ומופעים עם נתונים אמיתיים

### 3.1.3.4. טבלת ערכי החישנים בתיחסים ניהול מערך קרונות חכם

בטבלה 3.8 ניתן לראות את כל החישנים הנמצאים במערכת "ניהול מערך קרונות חכם".

עבור כל חישון בטבלה מוצג הערך אשר יכול להישלח במערכת, סוף השינוי שלו (כלומר מת' משנתה הערך שלו), המצב שלו והפעולה אשר מבוצעת (שליחת התראות אל מרכז הבקרה)

פעולה	מצב	סוף השינוי	ערך נשלח	חישון
עדכון מרכז שליטה על מיקום חדש	שינוי מיקום	new xyz	xyz	GPS
	מהירות נמדדת - V	פקודת התחלת נסעה		
דיווח למרכז שליטה על מהירות הקוון	קרון ב מהירות הרצויה	V < 5km/h	V	מהירות
דיווח למרכז שליטה על מהירות הקוון	מהירות נמוכה או גובהה מהרצוי	5km/h < V < 15km/h	V	
דיווח למרכז שליטה על מהירות הקוון	מהירות נמוכה או גובהה ומעבר מס' עצירת החירום	V > 15km/h	V	

טבלה 3.8: ניהול מערך קרונות חכם - טבלת ערכי חישנים

### 3.1.4. מידול וסימולציה באמצעות System Composer

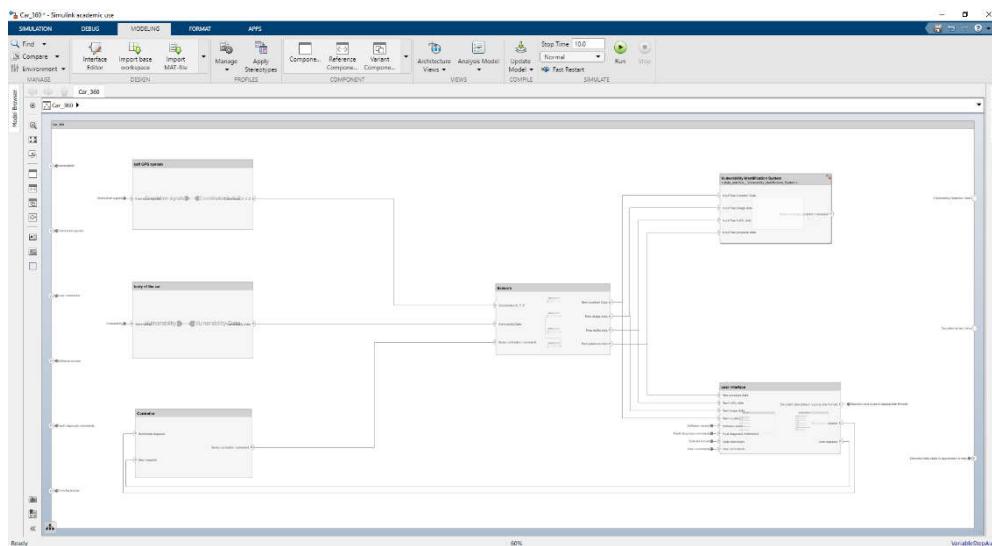
- פירוט המודל של כל המערכת נמצא בדף ג-1

במחקר ישימות הכליל System Composer בוצע מידול עבור 2 דוגמאות:

1. ניטור פגיעה ברכיב 360
2. ניהול מערך קرونות חכם

כאשר אנו משתמשים על ה- System Composer ניתן לראות הארכיטקטורה ואת היחסיות השונות בארכיטקטורה

באירור 3.1 ניתן לראות דוגמה לארכיטקטורה הכללית

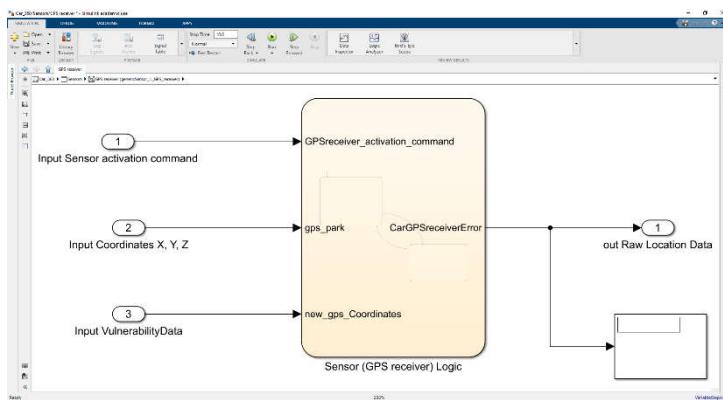


איך 3.1 - ארכיטקטורה כללית

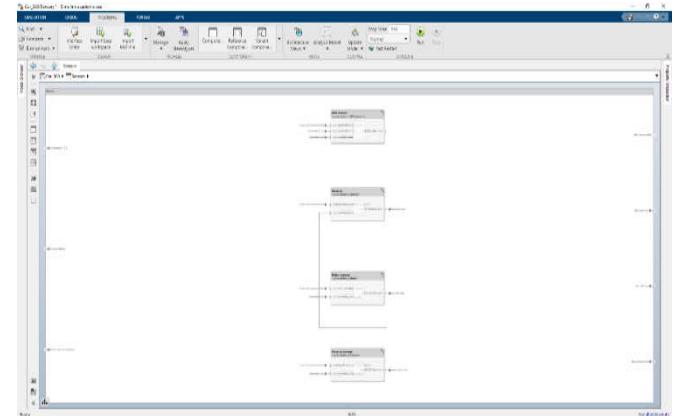
כאשר ניתן להיכנס אל כל ישות, אשר יכולה להכיל עוד ישות, וב就这样 ניתן לצולע עד הרמה בה החלטנו לעצור ולישם את הפעולות שאנו רוצים לבצע.

לדוגמה: אם נרצה לראות את פעולות אחד החישנים ניתן להיכנס אל תוך היחסום הנזכר "Sensors", לאחר מכן לבחור את החישון הרצוי ולהיכנס אליו.

באיורים 3.2 ו- 3.3 ניתן לראות דוגמה לסקימת החישנים (איור 3.2) ו- state flow להישן GPS באופן ספציפי (איור 3.3) אשר מאפשר להגדיר את לוגיקת/התנהגות החישן



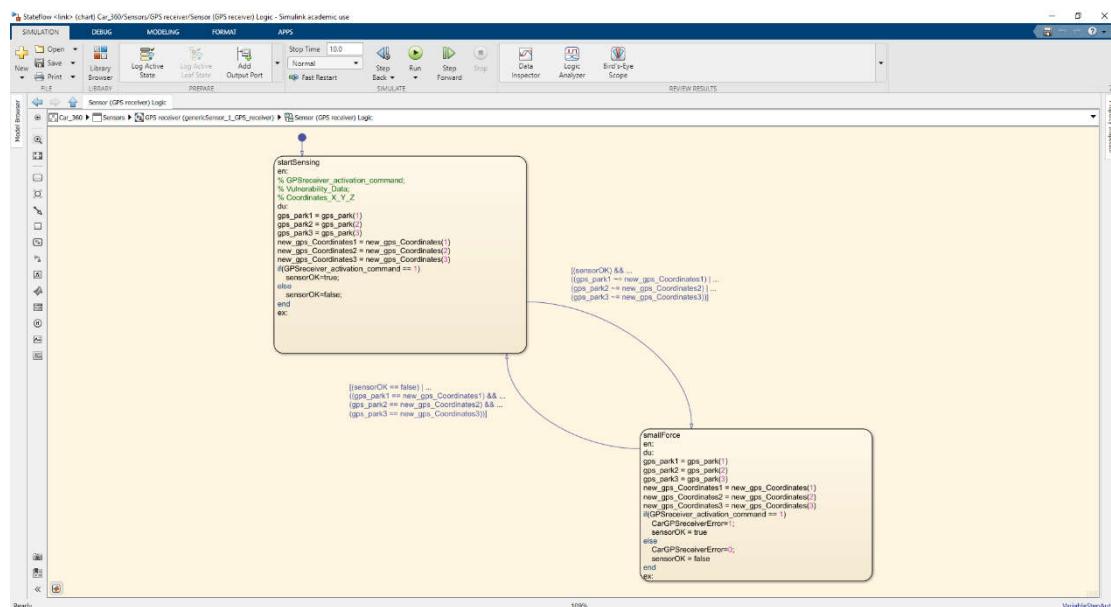
איור 3.3 - סקימת חישן GPS



איור 3.2 - מודל ארכיטקטורת החישנים System Composer

למעשה, כתוב ניתן לראות בבירור את כל הכניסות והיציאות של החישן, על מנת לראות את הפעולות המבוצעות ניתן להכנס אל תוך ה-state flow (ה-ישות הצהובה), ובעצם לראות את הפעולות אותן מבצע החישן ובאיזה תנאים.

#### דוגמה ל chisen בודד ניתן לראות באיור 3.4



איור 3.4 Stateflow של חישן בודד System Composer - State flow

החיישן שלפנינו הוא חיישן ה-GPS, וכפי שניתן לראות הוא משווה בין הקואורדינטות שננעו במערכת לבין הקואורדינטות הנוכחיות וזאת על מנת לבדוק האם קיימים שינויים במקומות. אם וכאש יש שינוי החיישן ישלח את הערך "1", אם אין שינוי מצד החיישן הוא ישלח את הערך "0".

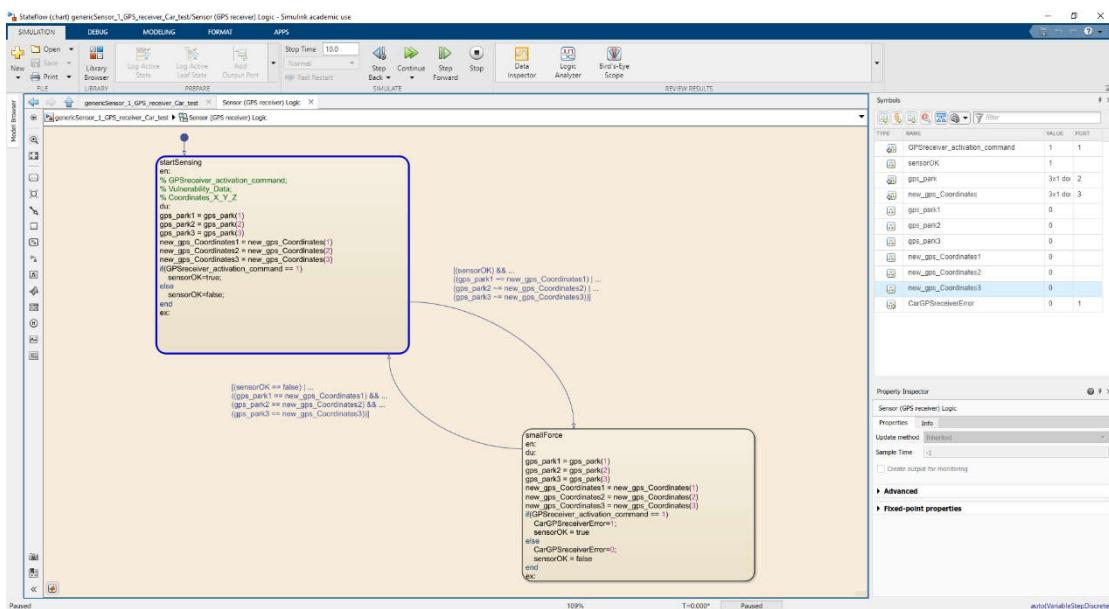
תוצאות החישן, על פि הארכיטקטורה, מגיע אל הישות הנקראת "Vulnerability Identification System" ובעצם על מנת לראות מה הפעולות אשר היא מבצעת יש לחזור על התהליך שהוסבר לעת.

כמו כן, המודל מורכב לא רק מישויות אלא גם מהקישורות השונות, profiles and stereotypes אשר מייצגים את הפורטים השונים (כניות ויציאות) ואת מה שמועבר באותו התוך עצמו. בנוסף ישנים קישורים אל דרישות המערכת, דרישות כליליות ודרישות מישיות או תתי ישוויות ספציפיות, אל דרישות אלה ניתן להגיע בלחיצת כפתור מתוך כלי ה-MATLAB.

מבחןת הסימולציה עצמה, ניתן לבצע את הסימולציה במספר צורות, תוצאה מידית / שלב אחר שלב תוך ניתוח המידע בזמן אמיתי. כאשר אנו מדברים על סימולציה שלב אחר שלב, ניתן לראות את הנתונים בכל שלב, את המעברים השונים, ההתנוויות ועוד, דבר אשר עוזר בניתוח והבנה عمוקה של הסימולציה, בעקבות סימולציות אלה התנוויות ופעולות המערכת השתנו מספר פעמים עד אשר הגיעו למצב הרצוי (ללא באגים)

## שלב 1 : מצב התחלתי

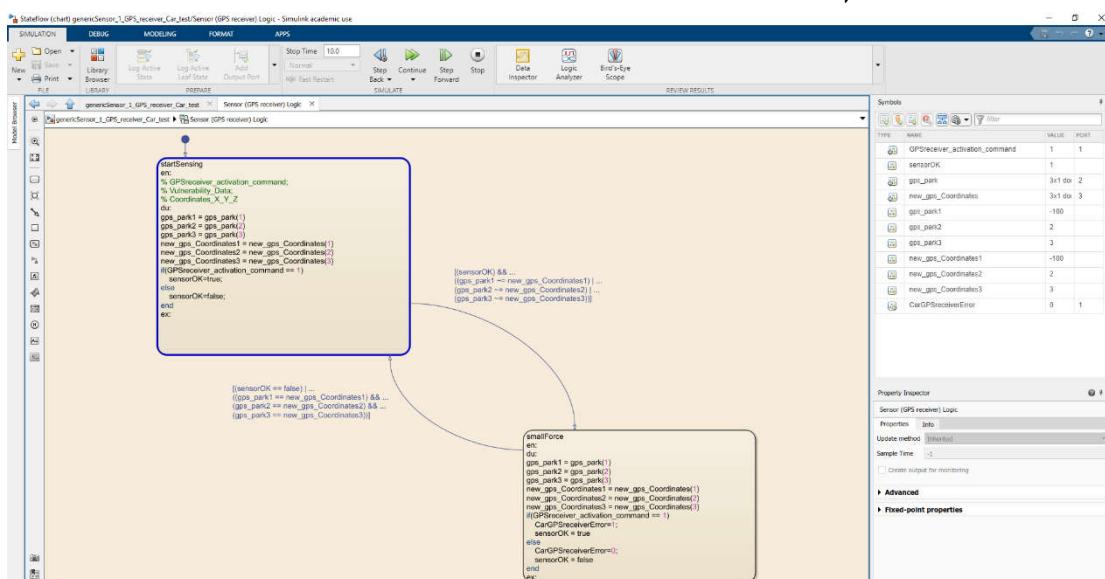
### באיור 4.5 ניתן לראות דוגמה למצב התחלתי



איור 3.5 – סימולציה - מצב התחלתי

## שלב 2: הנ吐ונים נשארו זהים ולכן אנו נשארים באותו מקום

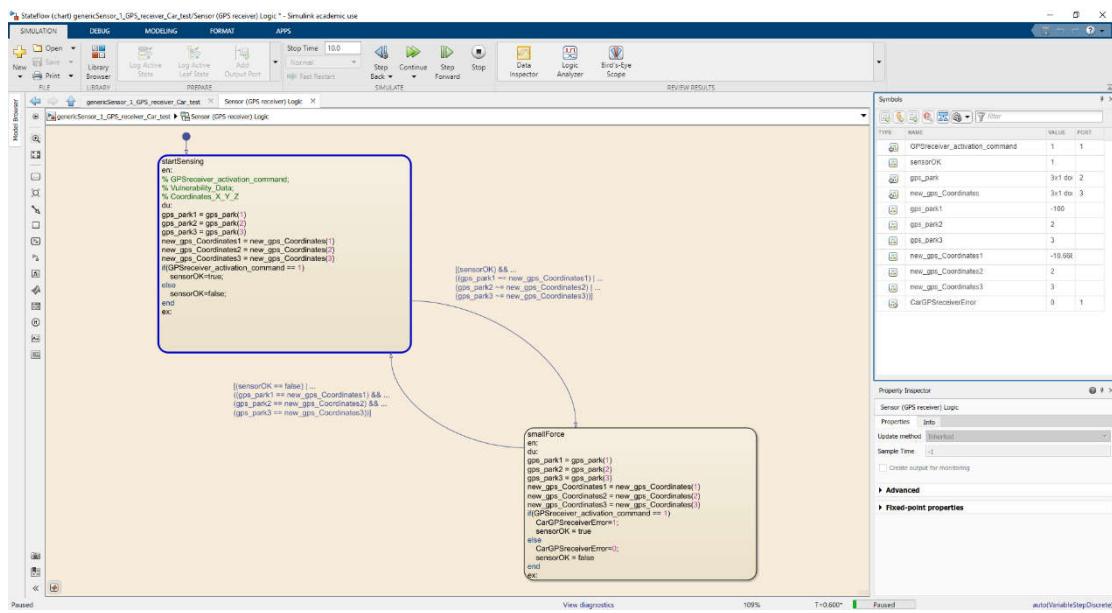
### באיור 3.6 ניתן לראות דוגמה לסימולציה – השארות נתוניים זהים



איור 3.6 – סימולציה - נתוניים זהים

שלב 3: הנזונים שונים, ניתן לראות את השינוי ובשלב הבא נוכל לראות ממש את המעבר, הבדיקה הזאת בקלט השינוי בשלב אחד והמעבר בשלב אחריו מKENA ידע רב על חקירת מאורע מסוים ובהנה עמוקה מאוד במתוך

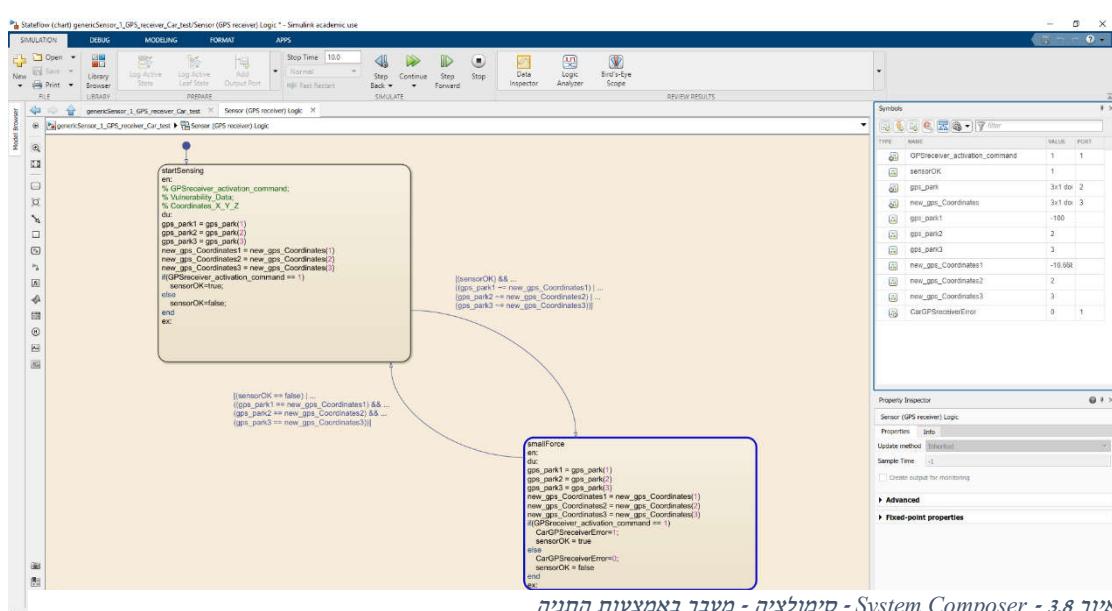
### באיור 3.7 ניתן לראות דוגמה לסימולציה – שינוי בנתונים



איור 3.7 - סימולציה - שינוי בנתונים

### שלב 4: מעבר באמצעות התניה

### באיור 3.8 ניתן לראות דוגמה לסימולציה – מעבר באמצעות התניה



איור 3.8 - סימולציה - מעבר באמצעות התניה

### 3.1.5. מידול באמצעות Enterprise Architect

- פירוט של כלל המערכת נמצא בדף ג-2

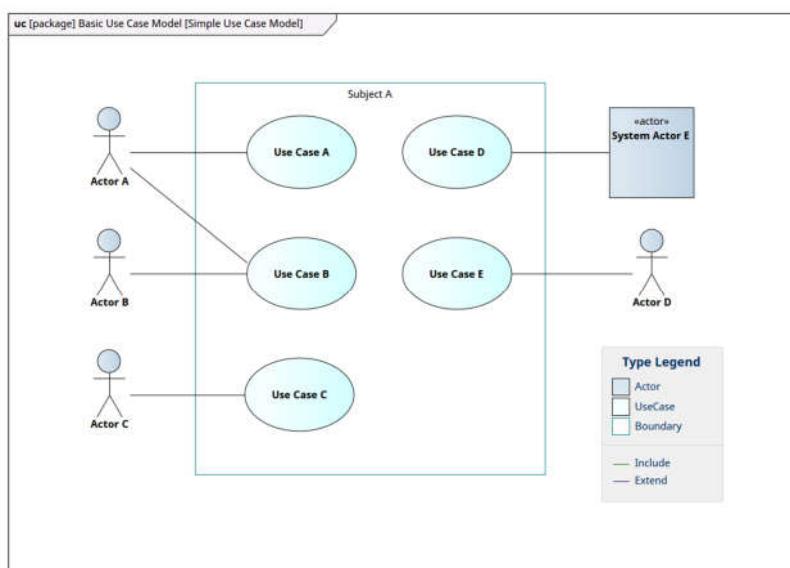
במחקר הנ"ל באמצעות הכליםEnterprise Architect בוצע מידול עבור 2 דוגמאות:

1. ניטור פגיעה ברכיב 360
2. ניהול מערך קرونוט חכם

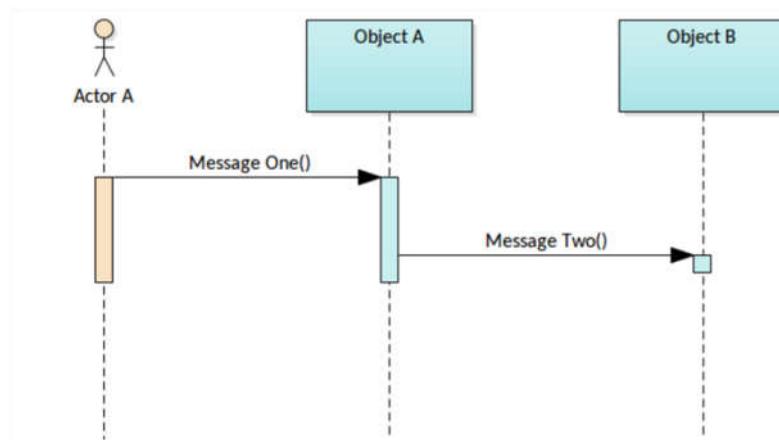
כאשר אנו עובדים עם Enterprise Architect אנו עובדים בשפת SysML ואת בניית המערכת נבצע על פיה שפה זו.

בכלי ה- Enterprise Architect נוכל לבצע את  
ה- Requirement Diagram, (Use Case Diagram) UCD  
וה- BD, (Sequence Diagrams) SD בהם  
השתמשנו ב- System Composer

באיורים 3.9 ו- 3.10 ניתן לראות דוגמה ל- UCD ו- SC ב- Enterprise Architect



*(Enterprise Architect - UCD (Use Case Diagram :3.9*

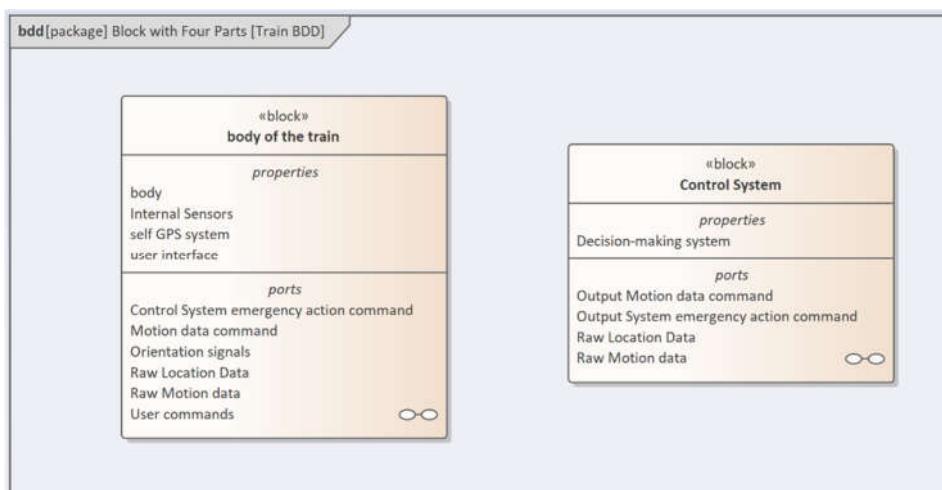


איור 3.10 Enterprise Architect - SD (Sequence Diagrams :3.10

דרך המידול:

ראשית נבנה BDD שיכיל את כלל ה"בלוקים" הקיימים במערכת.

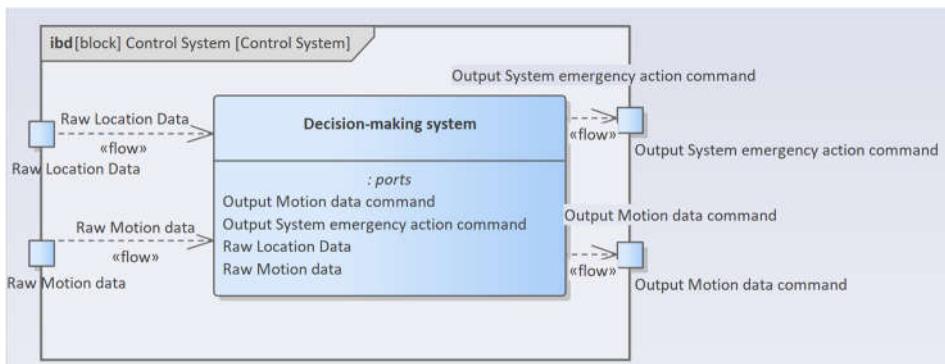
באיור 3.11 ניתן לראות דוגמה ל- BDD של כלל המערכת



איור 3.11 Enterprise Architect - BDD של כלל המערכת

לכל "בלוק" במערכת נבנה IBD אשר יוכל פירוט לגבי כלל תת-המערכות שבו והקשרו ביניהם.

באIOR 3.12 ניתן לראות דוגמה של IBD עבור בלוק בודד

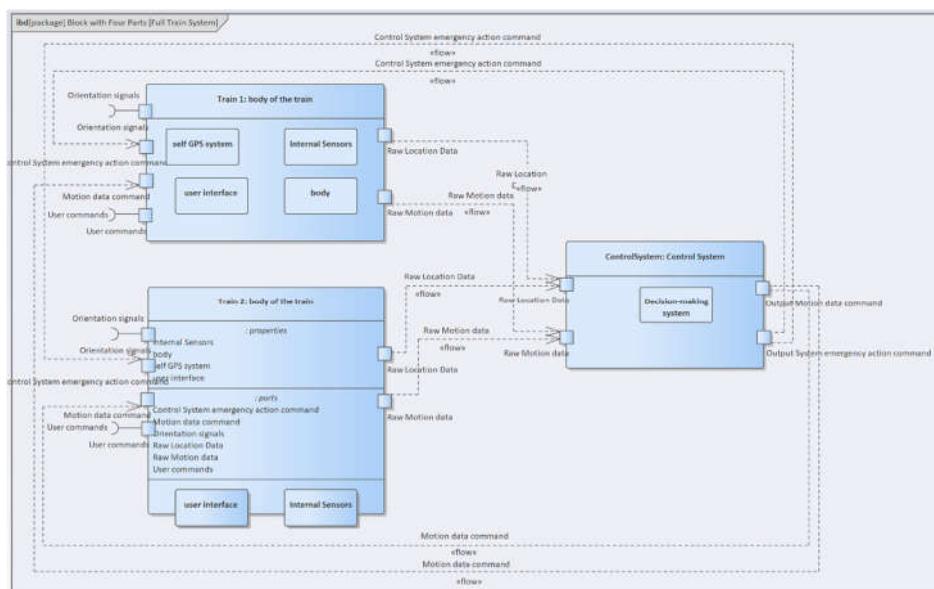


איור 3.12 Enterprise Architect - IBD של בלוק בודד

בשלב האחרון נבנה IBD לכל המערכת שיכיל את כל האלמנטים הכלולים במערכת, ע"י שימוש ב- IBD של כל אלמנט, והקשרו ביניהם.

כאשר אלמנט החוזר על עצמו מספר פעמים לא נדרש לבנות פעם נוספת אלא משתמש באותו IBD של האלמנט שבנוינו בהתחלה.

### באיור 3.13 ניתן לראות דוגמה ל-IBD של כלל המערכת

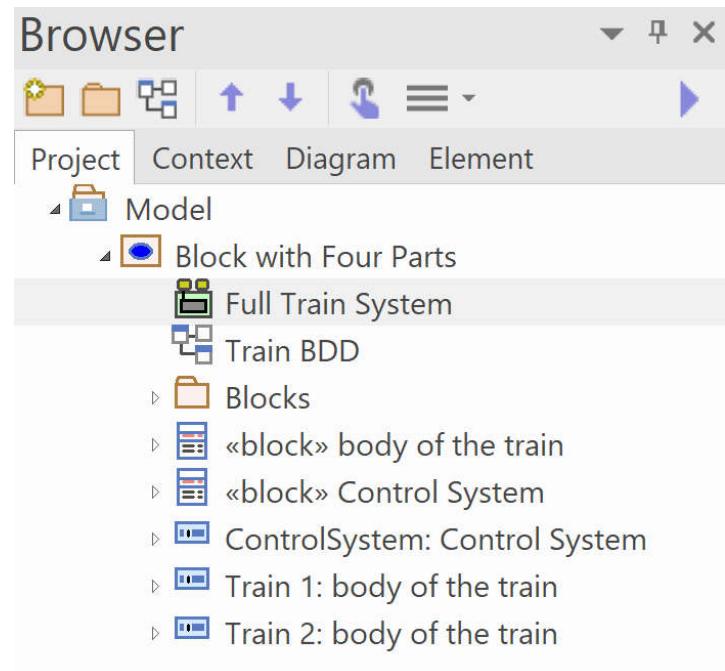


אילן 3.13: IBD של כלל המערכת Enterprise Architect - IBD

בתמונה האחורונה ניתן לראות שהשתמשנו בעמיהם ב- IBD של גוף הקרון (זאת רק דוגמא להמחשה, ניתן להשתמש בכמה מופעים שנדרש במידת הצורך) ופעם אחת ב- IBD של מרכז השליטה כי הוא רכיב אחד.

בעבודה עם כלי ה- Enterprise Architect יש לנו סרגל התמצאות באמצעותו אנחנו יכולים לעبور בין החלקים השונים של המערכת, כאשר מה- BDD ניתן לעبور ל- IBD גם ע"י לחיצה כפולה על הבלוק.

באיור 3.14 ניתן לראות דוגמה לסדרת התוצאות

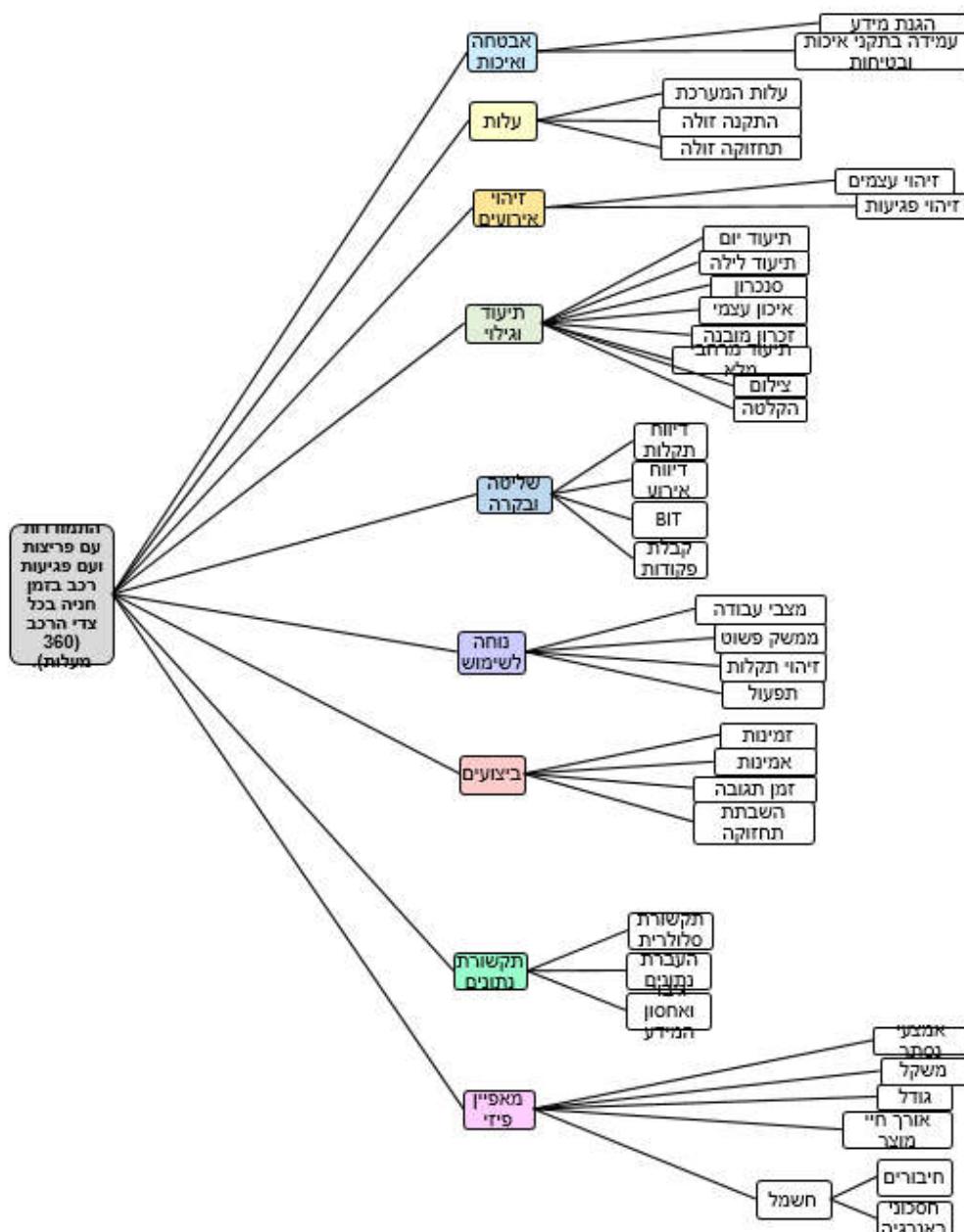


אייר 3.14 - סדרת התוצאות - Enterprise Architect :3.14

### 3.2. עץ צרכים

באירור 3.15 להלן מוצג עץ הצרכים אשר נבנה עבור דוגמת "nitro פגיעה ברכב 360" על מנת להציג את הבסיס המחשבתי ההתחלתי שמננו לבנה המודל כפי שנעשה בפרויקט אמיתי.

- עבור הדוגמה "ניהול מערך קרונות חכם" לא קיימים עץ צרכים וזאת מכיוון שמדובר על קרון אחד פשוט אשר מוכפל במספר רב של פעמים لكن לא היה צורך להגדיר עץ צרכים מפורט



אייר 3.15: עץ צרכים

### 3.3. טבלת דרישות

להלן טבלאות הדרישות, דרישות אלו הן מצד אחד נקודת ההתחלה ההנדסית לבניית המודלים, ומצד שני משמשים בסיס לבניית תוכנית בדיקות בבנייה אמיתית של מערכת ובמקרה של המחקר שלנו שישמו בסיס להגדרת הערכיים בתרחישים ובמופעים ששימושם לביצוע סימולציה.

#### 3.3.1. טבלת דרישות - ניטור פגיעה ברכיב 360

בטבלה 3.9 להלן ניתן לראות את הדרישות עבור הדוגמא "ניטור פגעה ברכיב 360"

סוגיות בהבנה וסיכום	רמת האימוט	שיטת האימוט	סוג הדרישה	הדרישה המטויבת	מזהה הדרישת חדשה	הדרישה	מזהה הדרישת
<u>סוגיה:</u> ماובטחת מן מי ?	System	By Demo	Non-Functional (Privacy)	המערכת תעמוד בתקני אבטחת מידע ISO 27001	1.1	המערכת תהיה ماובטחת	1
<u>סוגיה:</u> ماובטחת מן מי ?	System	By Demo	Non-Functional (Privacy)	המערכת תהיה ماובטחת לפריצה מרוחק	1.2	היבטי סיביר	
<u>סוגיה:</u> כיצד היא תהיה בטיחותית?	System	By Demo	Non- Functional(RA MS)	המערכת תעמוד בטו תקן 60065 מ"י	2	המערכת תעמוד בתקני בטיחות של מכון התקנים	2
<u>סוגיה:</u> מה עלות הרכישה הקסימלית הקבילה?	System	by- inspection	Non-Technical	עלות המערכת ללקוח לא תעלה על 1000 ש"ח	3	המערכת תהיה מתאомת למחירים השוק	3

<u>סוגיה:</u> מה עלות התקנת המערכת המקסימלית הקבילה?	System	by-inspection	Non-Technical	מחיר התקנת המערכת לא עולה על 500 ש"ח	4	התקנת המערכת תהיה זולה	4
	System	by-testing	Non-Functional (Performance)	המערכת צריכה הספק של עד 1.2 וואט	5	המערכת תחסוך בצריכת אנרגיה	5
<u>סיכון:</u> התנגדות עצמתית ברכוב	System	by-testing	Non-Functional(RA MS)	המערכת תהיה בעלת זמינות של לפחות 98%	6	המערכת תהיה פעילה 24/7	6
<u>סוגיה:</u> עבור איזה ממשקים אנו מעוניינים שהמערכת שלנו תתמוך?	System	by-testing	Non-Functional-Environmental Conditions	המערכת תתחבר למתחם חשמלי של מctror הרכב	7	משמעותי חיבורו של	7
	System	by-analysis	Non-Functional (environmental)	עמידה בטמפרטורה עד 70	8	עמידה לתנאי סביבה	8
	System	by-inspection	Functional	המערכת תאפשר להגיע לכל אחת מהfonקציות הראשיות ללא יותר מחמשה קליקים	9.1	משמעותי למשתמש	9
	System	by-inspection	Functional	המערכת תציג הודעות לשימוש בשפות: עברית, אנגלית, סינית, יפנית - לפי בחירתו	9.2		

<u>סוגיה:</u> כיצד ניתן היה לגשת למידע?	System	by-testing	Functional	המערכת תבצע ביטוחוני ותדועה סטטוס	10.1	זיהוי תקלות	10
	System	by-Demo	Functional	המערכת תשמר את התקלות	10.2		
<u>סוגיה:</u> איזה סוג תקלות יוצגו למפעיל ואיזה מידע עובר הלאה.	System	by-testing	Functional	המערכת תציג את התקלות למפעיל במרקם בקרה	10.3		
<u>סוגיה:</u> מה המדריך למערכת אמינה? כמה זמן אנו מוכנים שיעבור בין תקלה לתקלה?	System	by-Demo	Non-Functional(RAMS)	MTBF של המערכת יהיה גבוה	11	אמינות המערכת תיכון ע"י מדדים מקובלים	11
	System	by-testing	Non-Functional(Performance)	במערכת יהיה ניתן לבחור בין רזולוציות צילום	12.1		
	System	by-testing	Non-Functional(Performance)	המערכת תתעד אירועים באזוריים ללא תאורה	12.2		
<u>סוגיה:</u> כמה מעילות זה רחבה?	System	by-testing	Non-Functional(Performance)	זרוית צילום רחבה	12.3	מצלמה איקוטית לכל מצב תאורה	12

<u>סוגיה: באיזה פורמט יש להציג את הזמן ? 12\24 איזה אзор זמן ?</u>	System	by-testing	Functional	המערכת תציג למשתמש זמן עדכני	13	הציג זמן	13
<u>סוגיה: איזה נתונים נסנכרן</u>	System	by-testing	Functional	המערכת תסנכרן נתונים עם בסיס נתונים	14.1	גיבוי והעברת נתונים	14
<u>סיכון: רשות לא זמינה g4</u>	System	by-testing	Functional	תüberות הנתונים תבוצע תחת רשות דור 4	14.2		
	System	by-testing	Non-Functional (Performance)	המערכת תאפשר רדיוס תיעוד של 5 מטרים מהרכיב	15.1	טוווח תיעוד	15
	System	by-testing	Non-Functional (Performance)	המערכת תאפשר תיעוד מלא במרחב - 360 מעלות	15.2	הצילום	
	System	by-testing	Non-Functional	המערכת תזהה פגיעה ברכיב עקב תאונה או פריצה	16	זיהוי פגיעה ברכיב	16
	System	by-inspection	Non-Functional (Physical Characteristics)	אמצעי התיעוד יהיה נסתר	17	נראות המצלמות	17

טבלה 3.9: ניטור פגיעה ברכיב 360 - טבלת דרישות

### 3.3.2. טבלת דרישות - ניהול מערכות קرونוט חכם

בטבלה 3.10 להלן ניתן לראות את הדרישות עבור הדוגמא "NEYTOR מערכות קرونוט חכם"

סוגיות בהבנה וסיכוםים	רמת האימונות	שיטת האימונות	סוג הדרישה	הדרישה המטוויבת	מזהה	הדרישה חדשה	הדרישה	מזהה
					הדרישה חדשה			
<b>סוגיה:</b> ماובטיחת מפנוי מי?	System	By Demo	Non-Functional ((Privacy	המערכת תעמוד בתקני אבטחת מידע ISO 27001	1.1	המערכת תהיה מאובטחת בהיבטי סייבר		1
<b>סוגיה:</b> ماובטיחת מפנוי מי?	System	By Demo	Non-Functional ((Privacy	המערכת תהיה מאובטחת לפירצה מרוחוק	1.2			
<b>סוגיה:</b> כיצד היא תהיה בטיחותית בת?	System	By Demo	Non-Functional (RAMS)	המערכת תעמוד בטו תקן 60065 מתק"י	2	המערכת תעמוד בתקני בטיחות של מכון התקנים		2
<b>סוגיה:</b> שימוש במערכות איכון על כל קרון או כל מסילה?	System	by-testing	Non-Functional (RAMS)	המערכת תתמוך בתקן GPS סטנדרטי	3	למערכת תהיה יכולת ניטור כלל הcronos		3

	System	by-testing	Non-Functional (Performance)	המערכת צריכה הספק של עד 1.2 וואט	4	המערכת תחסוך בצריכת אנרגיה	4
	System	by-testing	Non-Functional (RAMS)	המערכת תהיה בעלת זמינות של לפחות 98%	5	המערכת תהיה פעילה 24/7	5
<u>סוגיה:</u> עבור איזה ממשקים אנו מעונייני אם שהמערכת שלנו תתמוך?	System	by-testing	Non-Functional - Environmental Conditions	המערכת תתחבר למתחם חשמלי של הקرون	6	ממתקי חיבורו של חשמל	6
	System	by-analysis	Non-Functional (environmental)	עמידה בטמפרטורה עד 70 מעלות	7	עמידה לתנאי סביבה	7
	System	by-inspection	Functional	המערכת מאפשרת הגיע לכל אחת מהפונקציות הראשיות ללא יותר	8.1	ממתק נוח לשימוש	8

				מחמישה קליקים			
	System	by-inspection	Functional	המערכת תציג הודעות למשתמש בשבות: עברית, אנגלית, סינית, יפנית - לפי בחירהו	8.2		
<u>סוגיה:</u> כיצד ניתן יהיה LAGSHET למידע?	System	by-testing	Functional	המערכת תבצע בית מחזורי ותדועה סטטוס	9.1		
	System	by-Demo	Functional	המערכת תשמור את התקלות	9.2		
<u>סוגיה:</u> איזה סוג תקלות יוצגו לפעול ואיזה מידע עובד הלאה.	System	by-testing	Functional	המערכת תציג את התקלות לפעול במרקם בקירה	9.3		
	System			MTBF	10		10

זיהוי  
תקלות

9

<u>סוגיה:</u> מה המדר למערכת אמינה? כמה זמן אנו מוכנים שיעBOR בין תקלה لتקלת ?		by- Demo	Non- Functional (RAMS)	של המערכת יהיה גבוהה		אמינות המערכת תיכון ע"י מדדים מקובלים	
	System	by- testing	Functional	המערכת לא תהיה תלויה במצבה מלאה של הקרונות ותוכל לפעול עם חוסרים	11	המערכת תוכל לפעול גם ללא כל הקרונות	11
<u>סוגיה:</u> באיזה פורמט יש להציג את הזמן ? 12\24 איזה אזור זמן ?	System	by- testing	Functional	המערכת תציג למשתמש זמן עדכני	12	הציג זמן	12
<u>סוגיה:</u> איזה נתונים נסנכרן	System	by- testing	Functional	המערכת תסנכרן נתונים עם בסיס נתונים	13.1	גיבוי והעברת נתונים	13

<u>סיכון:</u> רשת לא זמינה g4	System	by- testing	Functional	תעבורה הנתונים תתבצע תחת רשת דור 4	13.2		
-------------------------------------	--------	----------------	------------	--	------	--	--

טבלה 3.10: ניהול מעדר קיוניות חכם - טבלת דרישות

#### 4. פרק 4 – תוצאות הממחקר – בדיקת מדדי הממחקר

כפי שהגדכנו בתחילת הממחקר ישנים 6 מדדים להשוואה, כאשר לכל אחד מן המדדים ישנו משקל שונה.

בטבלה 4.1 להלן ניתן לראות את המדדים אל מול המשקלים שלהם

משקל		מדד
20%	נוחיות ויזואלית	1
20%	ביצוע הגדרת חזורת / עדכון הגדרה	2
10%	IN/OUT – שינוי כניסה ויציאה מהמערכות השונות	3
10%	שינוי ארכיטקטורה באמצעות הzzle אלמנטים	4
10%	שפת מידול מוכרת לכל המשתתפים בפרויקט	5
30%	השפעה של שינויים באמצעות הפרויקט	6

טבלה 4.1: טבלת מדדים

משקלים המדדים נקבעו לאחר חשיבה عمיקה על רמת ההשפעה של המדד על הפרויקט, מתחילה ועד סוף.

כפי שניתן לראות, מדד ה"השפעה של שינויים באמצעות הפרויקט" קיבל את המשקל הגבוה ביותר והוא מכיוון שהוא המשפיע ביותר על הפרויקט כולם.

מיד אחריו המדדים "נוחיות ויזואלית" ו- "ביצוע הגדרת חזורת / עדכון הגדרה" אשר משפיעים גם הם על הפרויקט כולם, יכולותיו והעומקתו קיבלו את השקלול השני הגבוה ביותר.

שאר המדדים "IN/OUT – שינוי כניסה ויציאה מהמערכות השונות", "שינוי ארכיטקטורה באמצעות הzzle אלמנטים" ו- "שפת מידול מוכרת לכל המשתתפים בפרויקט" ניתנים למידה ומשפיעים בצורה מעט נמוכה יותר על הפרויקט ברמה הכלכלית.

לאחר עבודה רבה והתנסות עם המערכת ובחינתן ע"י למידת המערכת, System Composer, Enterprise Architect ו- Enterprise Architect, ביצוע ארכיטקטורות, בניית תרחישים, וביצוע הנדסת מערכת לשני פרויקטים מעשיים שונים (פיתוח פגיעה ברכב 360 ו- ניהול מערכות חכם),

**בסעיפים 4.1-4.5 יוצגו התוצאות שהתקבלו על פי חלוקה לפי הכלים השונים בהם השתמשנו במהלך הפרויקט ולפי המדרים איתם מדנו את יכולות הכלים. יש לציין במיוחד:**

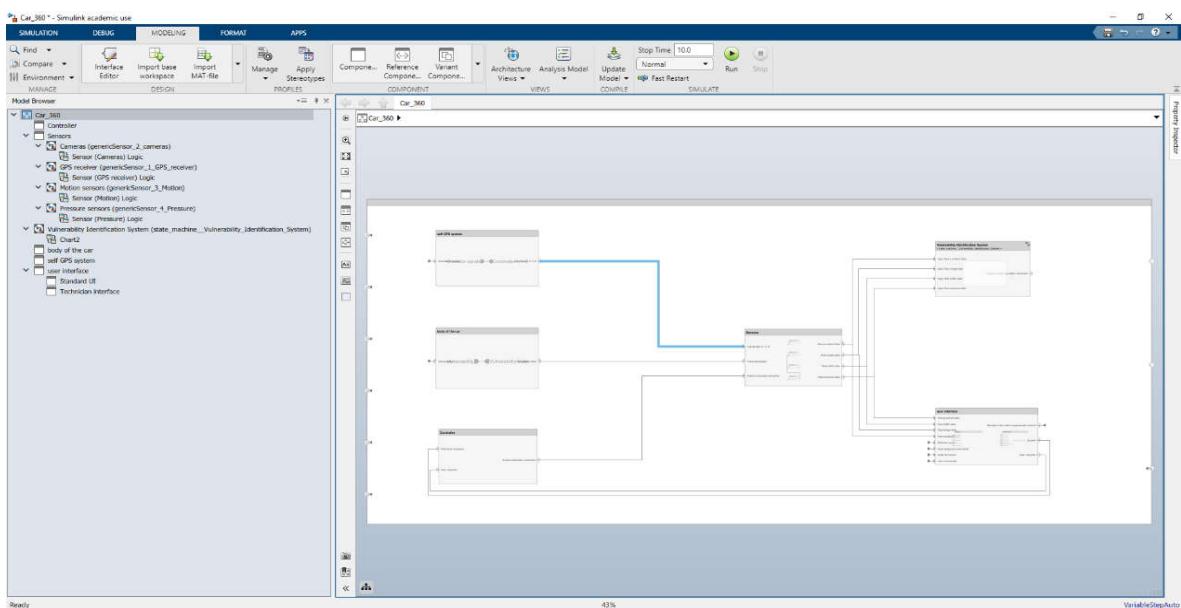
- **המודלים עצמם והנדסת המערכת** (כגון: דרישות, עץ צרכים, SD, וכו') נבנו באופן מעשי
- **בכלי System Composer**, הכלי החדש בשוק, האפשרות למידול פיזיקלי ולוגי באמצעות סט כלים שכחיהם בתעשייה (MATLAB, Simulink, State flow וכו')
- **בוצעה סימולציה מלאה** (ידנית ואוטומטית) סיכום הממצאים בנושא הסימולציה מוצגים בסעיף נפרד להלן, סעיף 4.5 וזאת עקב חשיבות הנושא

## System Composer .4.1

### 4.1.1. מדריך 1 - נוחיות ויזואלית:

מבחן ויזואלית, משק המשמש נוח מאד, ברור וקריא, ניתן לתת שמות מובנים ומכוונים, ישנו שם מייצג לכל "רכיב", כאשר כל מרכיבי המערכת מקושרים ביניהם באמצעות קישור אשר מ-2 צדדיו בעל שם זהה, על מנת להזות את הקישורויות בין 2 מרכיבים וזאת בנוסף לאפשרות להזוז על הקישור, אשר בעת הלחיצה מסומן בצע בולט. כמו כן, ככל שצוללים פנימה, ישנו ניתוב היררכי בראש העמוד אשר מייצג את ההיררכיה בה התקדמותו, באמצעות נתיב זה ניתן לחזור מכל מקום אחורה אל כל מקום. כמו כן ישנו תפריט אשר מציג את כל המרכיבים במערכת, דרכו ניתן לגשת ולהגיע אל כל רכיב באופן מיידי, תפריט זה עוזר גם להבין את ההיררכיה בין הרכיבים השונות.

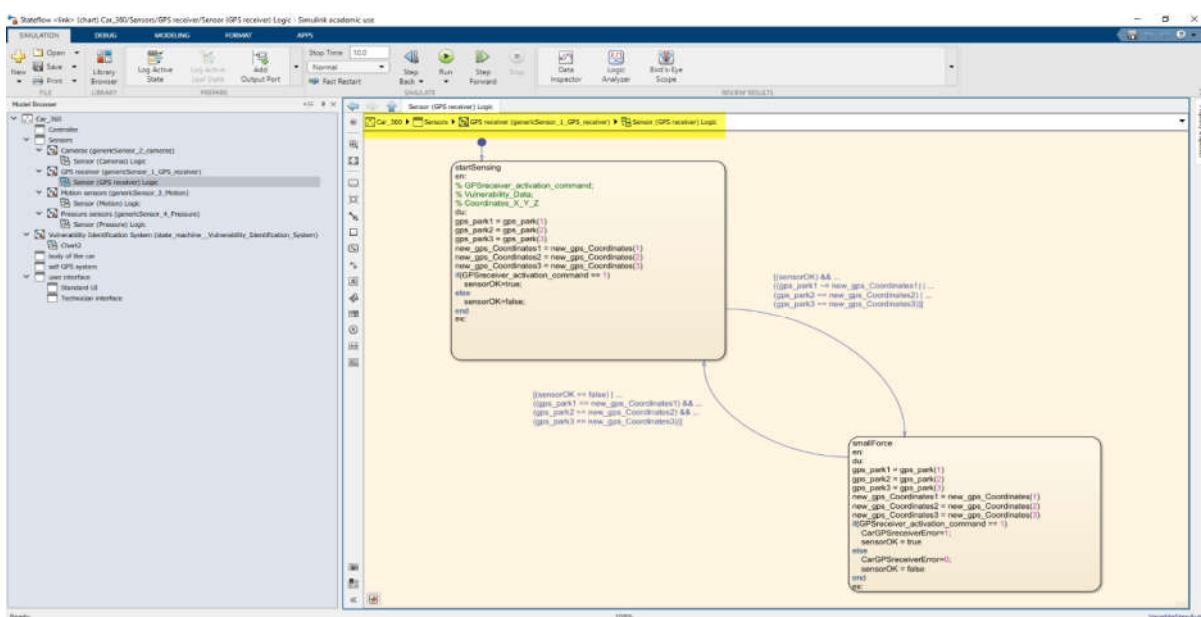
דוגמה לתפריט ניתוב בין רכיבים בארכיטקטורה הכללית ניתן לראות באירוע 4.1



אירוע 4.1 – חפירת ניתוב בין רכיבים בארכיטקטורה כללית – System Composer :4.1

באיור 4.2 ניתן לראות דוגמה ל-state flow בתוכו רכיב אשר מישם את לוגיקת/התנהגות

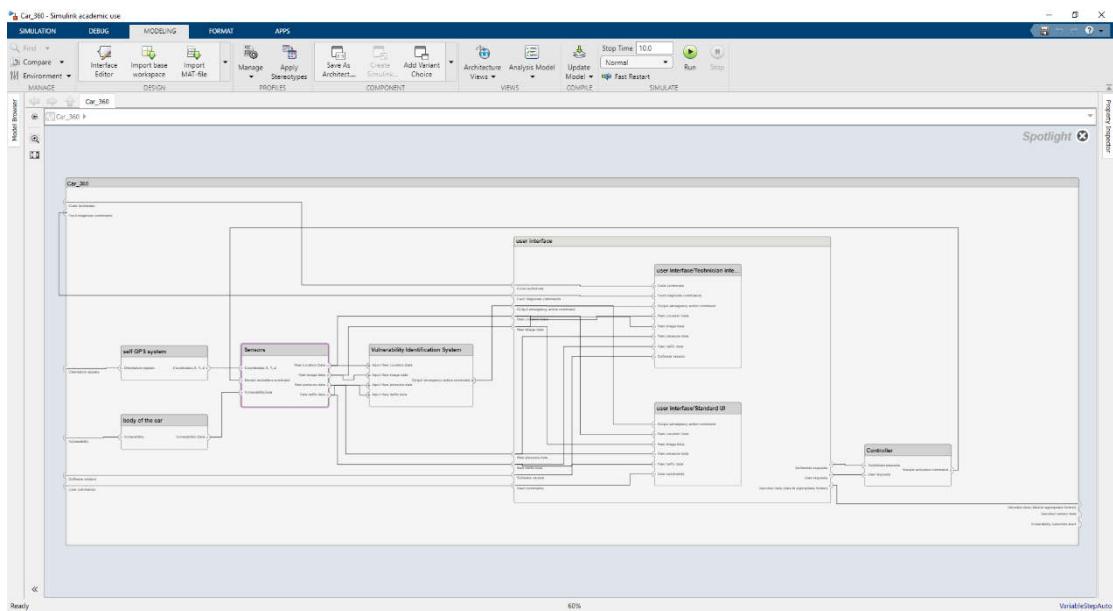
הרביב



## איד 4.2 System Composer - State Flow

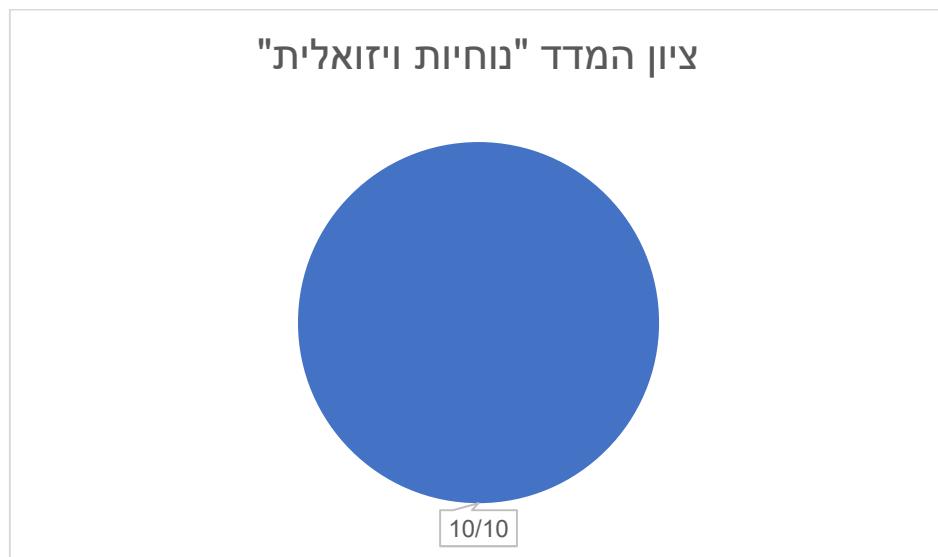
כמו כן, ישנה אפשרות לראות את השיקות והקשריות של כל רכיב במערכת וביצם להבין את ההיררכיה כולה עד הפרטים הקטנים ביותר,

**באיור 4.3 ניתן לראות דוגמה לכל הקשרים של מערכת החישבים**



איור 4.3 - חצצת קשרי הרליב

**באיור 4.4 ניתן לראות את ציון עבור המדד הנ"ל וחאת בהתאם ליכולות שצוינו לעיל**



איור 4.4 – ציון עבור מדריך נוחית ויזואלית – System Composer

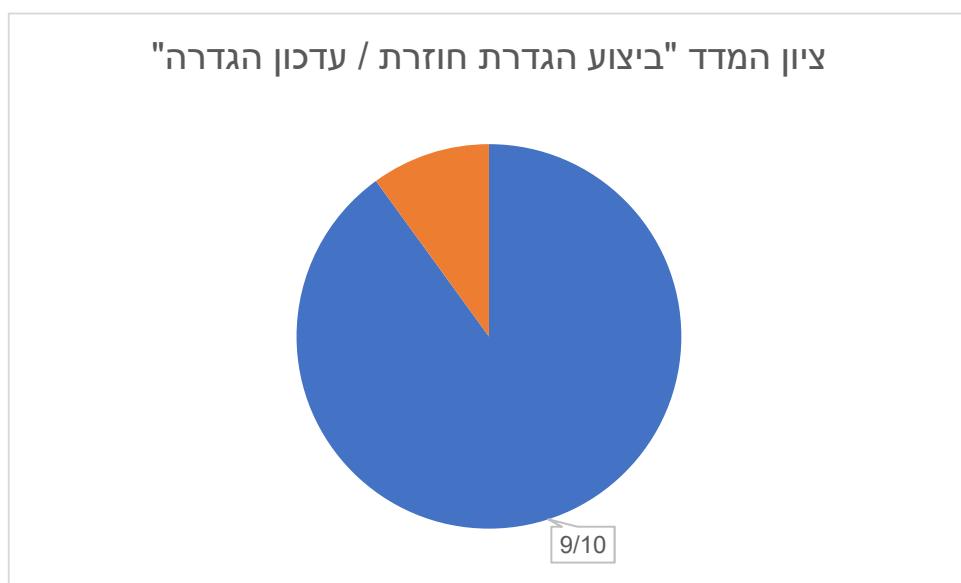
#### 4.1.2. מדריך 2 - ביצוע הגדרת חוזרת / עדכון הגדרה:

ביצוע הגדרות באופן חוזר היה די פשוט, כמעט הקישור עצמו של הרכיבים החדשניים שנוצרו באמצעות העתק-הדקק, אשר יצרה את אותם הרכיבים כולל תתי המערכות. כמו כן, ישנו הגדרות שניתן להפוך למשותפות, דבר אשר מקל את התהילין,

לעומת זאת, עדכון הגדרה עלול להיות מסובך מאוד, ישנו לא מעט הגדרות אשר תלויות האחת בשנייה, שינוי הגדרה יכול להשאיר לא מעט בעודה, כמו כן, שינוי ארכיטקטורה באמצע הפרויקט הוא דבר מסובך מאוד, גורר קשרים חדשים, מכונות שונות, ובעצם הגדרה מחדש.

- **הציפייה הייתה יכולה גיריה/ פעללה פשוטה – מדובר במדריך 5 ("שינויי ארכיטקטורה באמצעות הזוזת אלמנטים")**

#### איור 4.5 ניתן לראות את ציון עבור המדריך הנ"ל וחואת בהתאם ליכולות שצוינו לעיל



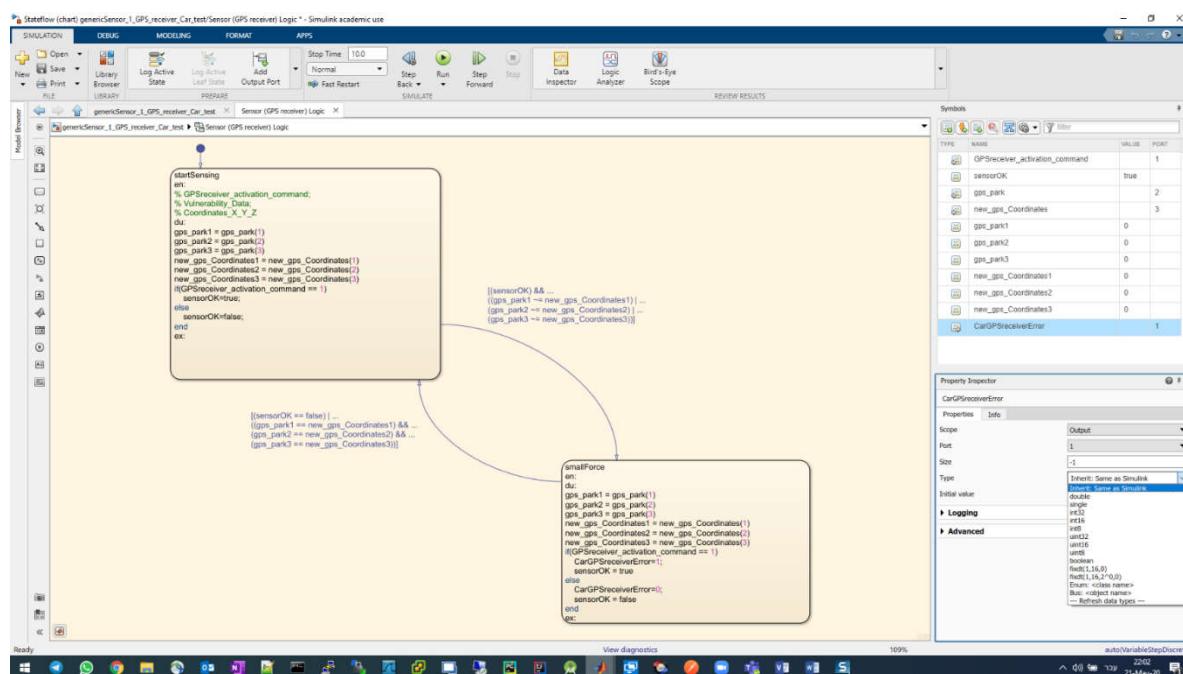
איור 4.5 – ציון עבור מדריך "ביצוע הגדרת חוזרת / עדכון הגדרה" – System Composer :4.5

#### 4.1.3. מודד 3 - IN\OUT - שינוי כניסה ויציאה מהמערכות השונות:

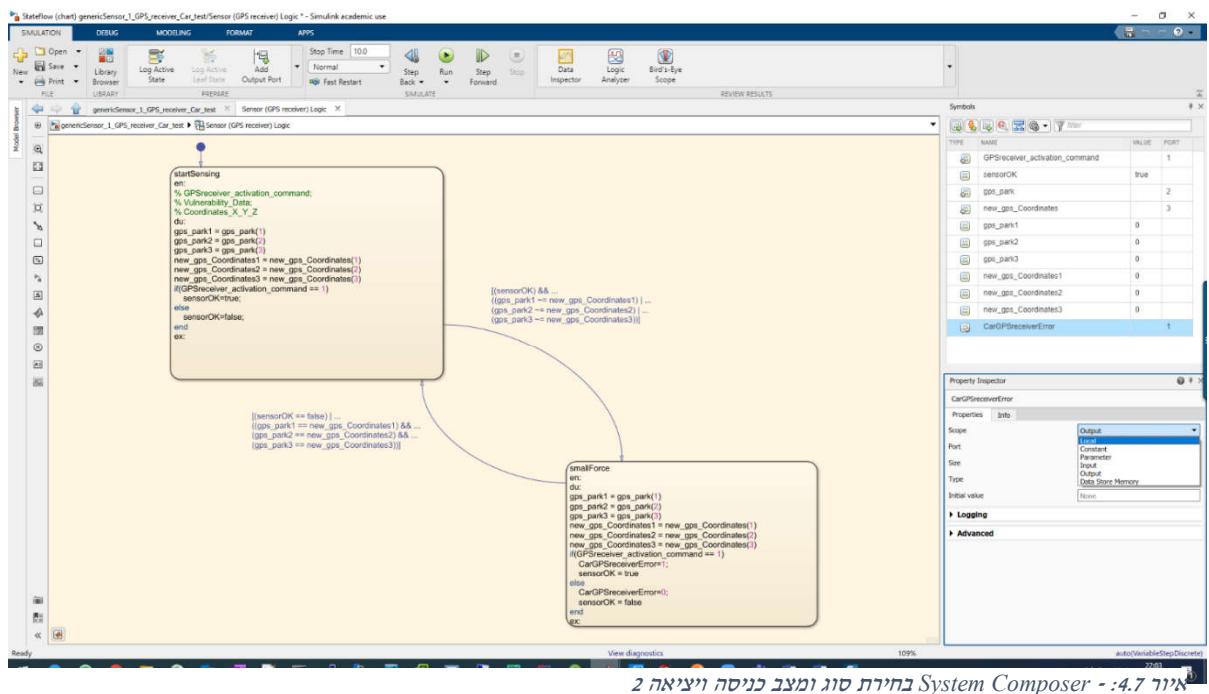
שינוי הכניסות והיציאות במערכת הוא נושא פשוט מאוד, הדבר מצריך את שינוי סוג הפורט (הכניסה או היציאה), בעצם באמצעות שינוי סוג הפורט ניתן לשנות את סוג הכניסות והיציאות, כמו כן שינוי כניסה ליציאה גם הוא פשוט ומתבצע באותה הדרך.

- היהתה צפיה שכאשר משנים כניסה/יציאה מסוימת יהיה ניתן לשנות את הכניסות והיציאות הקשורות אליה באופן אוטומטי וגורף

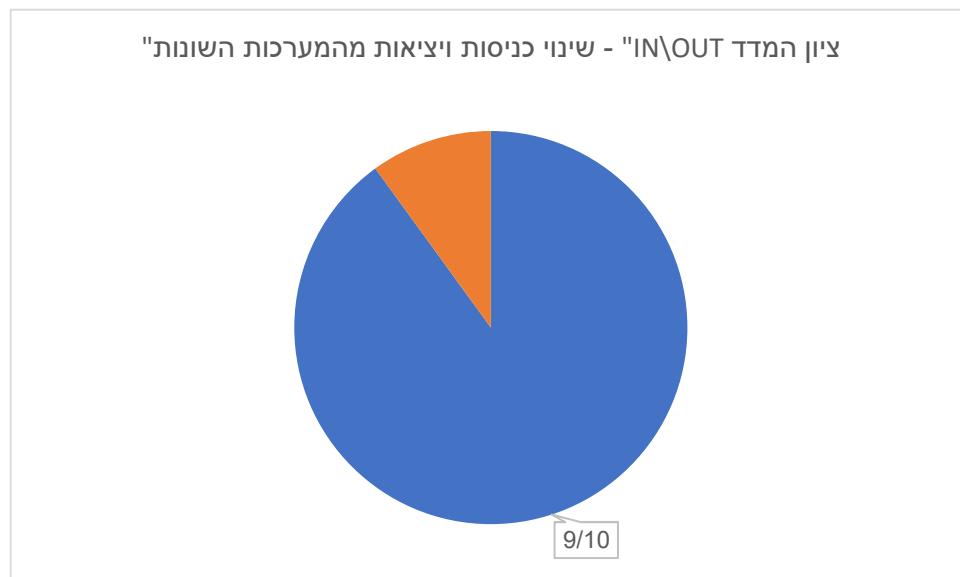
#### 4.6 ו- 4.7 ניתן לראות דוגמה לבחירת סוג הקישור (Input/Output)



אילו 4.6 - בחרית סוג ומצב כניסה ויצאה 1



**באיור 4.8 ניתן לראות את ציון עבורי המדד הנ"ל וזאת בהתאם לכליות שצוינו לעיל**

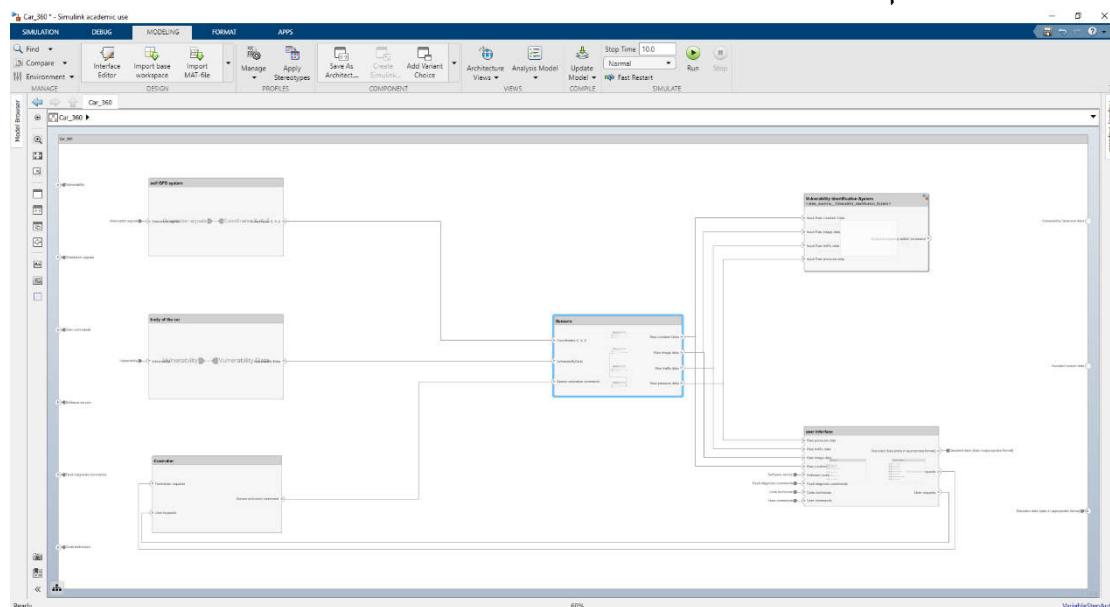


**איור 4.8:** System Composer – ציון עבור מד" – IN\OUT - שינוי כניסה ויציאה מהמערכות השונות

#### 4.1.4. מדריך 4 - שינוי ארכיטקטורה באמצעות הזנת אלמנטים:

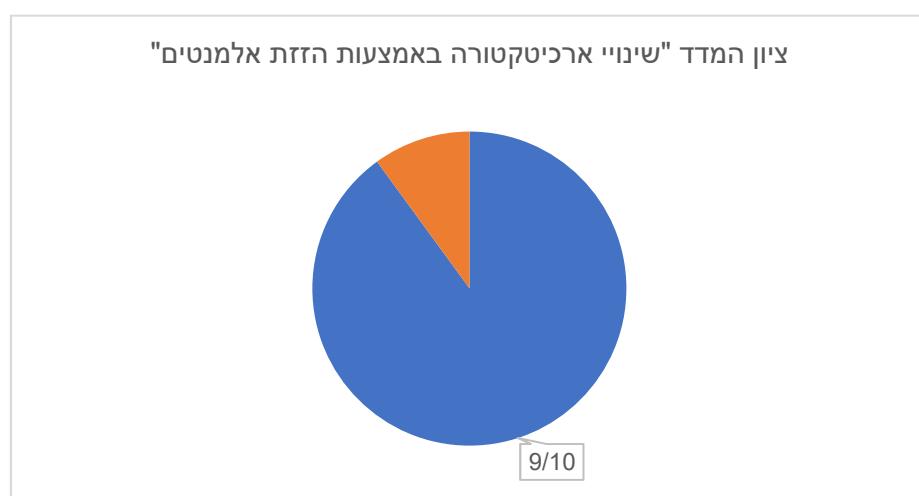
הזנת אלמנטים לטובת שינויים היא דבר פשוט. עם זאת, אין אפשרות לגרום לכך אל תוך רכיב אחר, יש לנתק אותה מכלל הקישורים שלו, להעתיק ולהדביק אותו במקום הרצוי ולאחר מכן יש לבצע שוב את כל הקישורים הנדרשים. כמו כן, יש להתאים כנישות ויציאות בהתאם לשינויים

**באיור 4.9 ניתן לראות דוגמה לבחירת רכיב**



**איור 4.9 - בחרת רכיב**

**באיור 4.10 ניתן לראות את ציון עבור המדריך וזאת בהתאם ליכולות שצוינו לעיל**



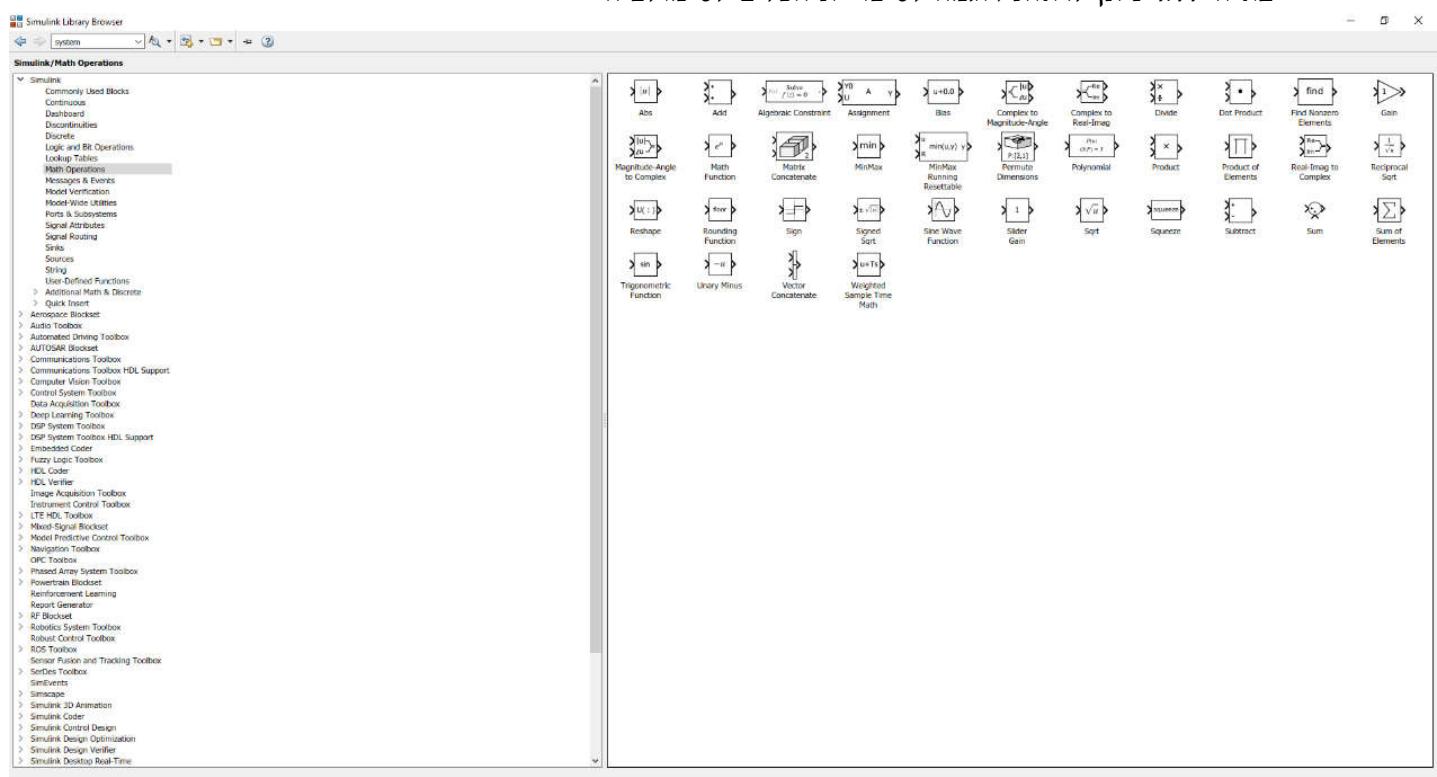
**איור 4.10 – ציון עבור מדריך "שינוי ארכיטקטורה באמצעות הזנת אלמנטים"**

#### 4.1.5. מדד 5 - שפת מידול מוכרת לכלל המשתתפים בפרויקט

בבנייה הארכיטקטורה הינה פשיטה יחסית אך בניית הסימולציה דורשת הבנה של המערכת בפרט עבור סימולציות מורכבות.

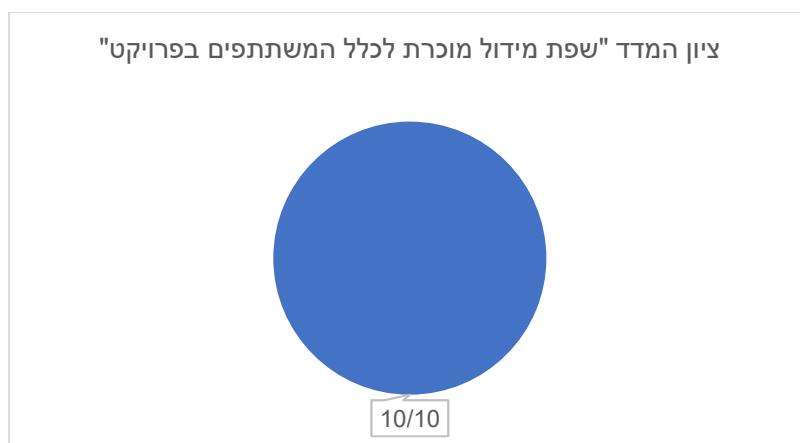
ישנם המונ אפשרויות, שילובים, ונדרשת הבנה מעמיקה אך עם זאת, ישנים מסמכים רבים באתר של MATLAB אשר מסבירים הכל, יש לציין שכמה מסמכים באתר יכולה לבלבול, אך לרוב מספקת את התשובות.

**איור 4.11 ניתן דוגמאות למספרית הכלים לסימולציה**



**איור 4.11 - ספירת כלים לסימולציה :4.11**

**באיור 4.12 ניתן לראות את ציון עבור המדריך וזאת בהתאם לכיקולות שצינו לעיל**

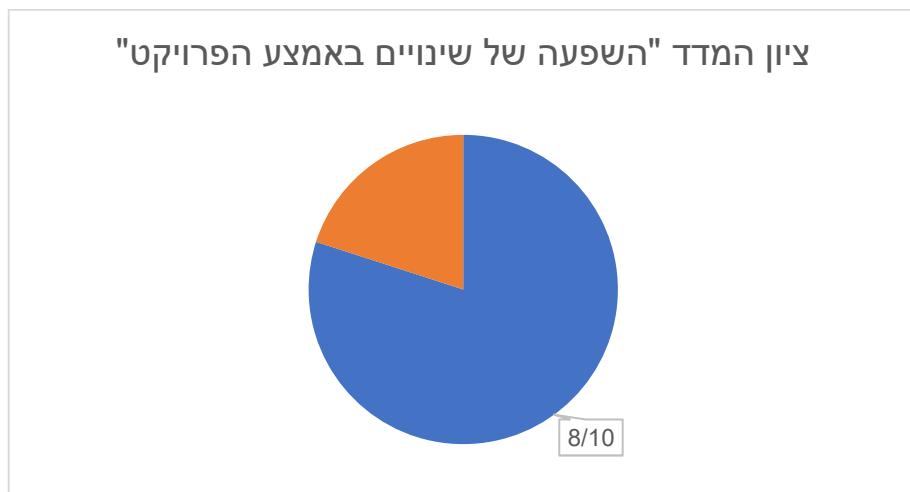


איור 4.12 – ציון עבור מדריך – שפות מידול מוכרת לכל המשתתפים בפרויקט

#### 4.1.6. מדריך 6 - השפעה של שינויים באמצעות הפרויקט:

- השפעה של שינויים באמצעות הפרויקט היא גדולה ורחבה ייְקָר, לאחר שבוצעו הרבות הגדירות, קישורים בין הרכיבים ו קישור בין השכבות השונות (state flow and System Composer) הדבר בכל הנראה יצריך שינוי מסיבי של כל האלמנטים.
- האלמנטים שככל הנראה נctrיך לשינוי ולעדוך הם: כניסה ויציאה, קישורים, פורטים, הוצאות רכיבים, התאמה של state flow אם עדין רלוונטי, אם אינו רלוונטי יש ליצור חדש, כמו כן, כפי שקרה בפרויקט שלנו, התחלנו עם כניסה מסוג Boolean ולבסוף נאלצנו לשינוי לסוג וקטור. השינוי הדרוש המון עבודה בכל רכיבי המערכת והיה לא פשוט.
- כמו כן, לאחר שינוי כזה ישנו צורך בבדיקות מחדש, ביצוע סימולציות מחדש, בדיקה וניתוח של התוצאות וכו'. דבר זה יכול בעצם להחזיר אותנו לנקודת ההתחלה,
- יש לציין שהפעולה נחשבת מסובכת אך פשוטה יותר וזאת בהשוואה אל כלים אחרים שנבדקו במהלך המחקר.

#### באיור 4.13 ניתן לראות את ציון עבור המדריך וזו בהתאם ליכולות שצוינו לעיל



אייר 4.13 – ציון עבור מדריך "השפעה של שינויים באמצעות הפרויקט" – System Composer

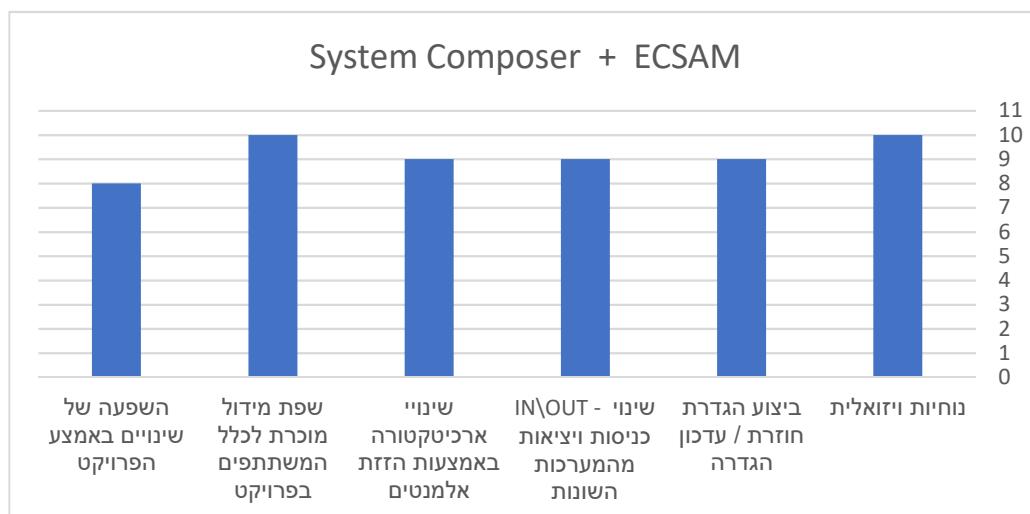
#### 4.1.7 יתרונות ל- System Composer :

- תצוגה ויזואלית ברורה
- מעבר בצורה פשוטה ובצורה מיידית בין הרכיבים השונים
- הוספה והסרה של רכיבים בצורה פשוטה ומהירה
- קישור פשוט בין הרכיבים השונים
- קבלת נתונים על כל קישור (פרמטרים, פרוטוקול, ירושות במידה וקיימות וכו')
- קישור מסמלי הדרישות/חלקים מהמסמך לכל רכיב במערכת
- ביצוע סימולציות בצורה פשוטה תוך קבלת נתונים על כל שלב ושלב
- שכפול אלמנטים גדולים ומורכבים (פרויקט ניהול מערכת חכם),
- יכולת שיתוף הפרויקט באופן פשוט

#### 4.1.8 חסרונות ל- System Composer :

- שינוי ארכיטקטורה עלול להיות מסובך מאוד
- אין יכולת להזיא מסמך מסכם ברמת עליון
- אין יכולת להזיא מסמך מסכם/דוח עבור כלל רכיבי המערכת
- ממשק WEB לא נוח עבור שיתוף ארכיטקטורת הפרויקט למשתמש ללא MATLAB

גרף המתאר את ציון המדדים ניתן לראות באירור 4.14



אייר 4.14 - גרף ציוני מדדים - System Composer + ECSAM

**שקלול הציון עבור הכלים System Composer בהתאם לחלוקת המדדים הוא: 9/10**

### 4.2.1. מדריך 1 - נוחיות ויזואלית:

מבחן ויזואלית, ממשק המשמש נוח, ברור ומסודר לפי כללי שפת המידול SysML אשר ע"פ עקרונותיה מתבצע המידול בכל אחד. ישנה אפשרות להגדיל ולהקטין חלונות ע"פ חסיבותם בשלב זה או אחר של העבודה הכללי.

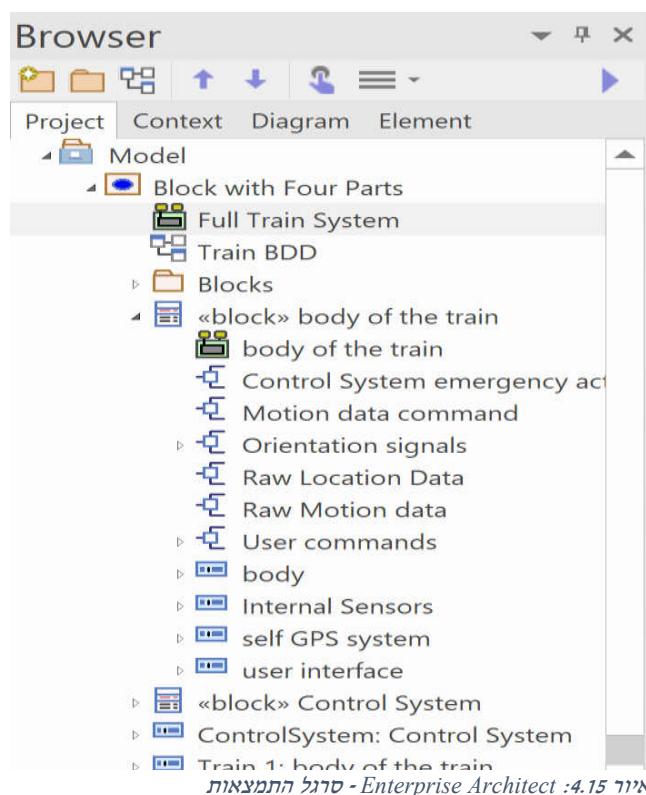
בצד שמאל ישנו סרגל התמצאות בו ניתן לראות את כל המרכיבים שבנו עד כה ואנו ניתן לגורור ולהשתמש בהם במבנה של מסמכים נוספים. בתמונה ניתן לראות את היררכיה:

IBD לכל המערכת, מתחתיו "בלוקים" אשר ניתנים לפיתחה ובهم נמצא את ה- IBD שלהם.

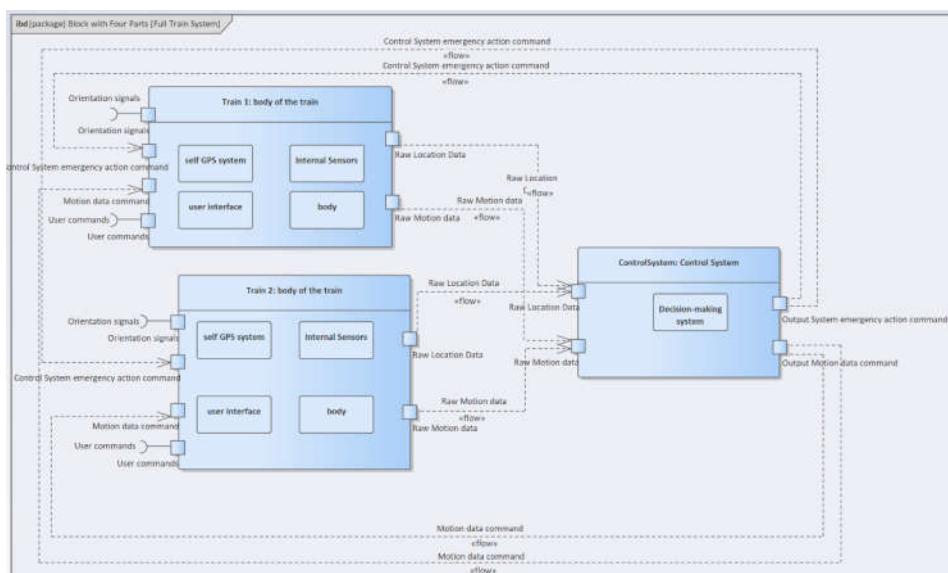
שלושת האלמנטים בכחול למטה הם האלמנטים בהם שהשתמשנו ב- IBD של כל המערכת.

\*שמות האלמנטים זהים לשמות הבלוקים (שהופיעו ב- BDD והוא להם IBD פרטני) מהם נלקחו וכן ניתן להזמין מהם נלקחו ל- IBD הכללי.

**באирור 4.15 ניתן לראות דוגמה לסדר התמצאות**



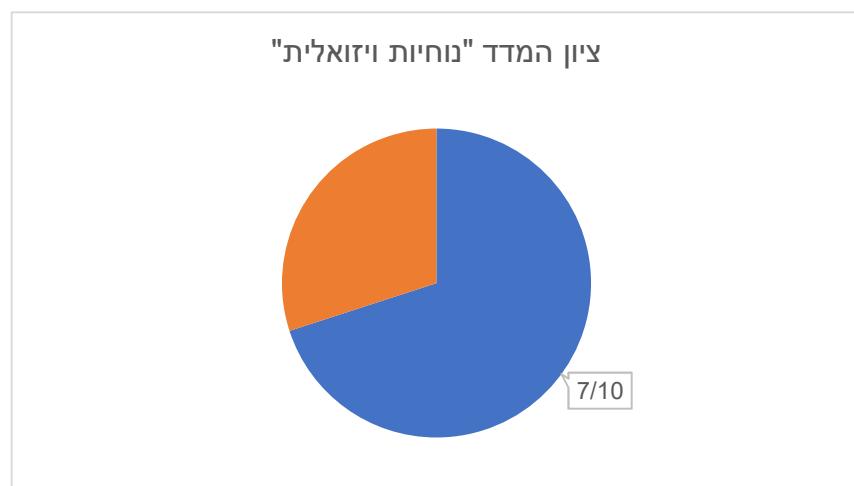
באיור 4.16 ניתן לראות דוגמה ל-IBD של כלל המערכת



#### **איור 4.16 Enterprise Architect - IBD :**

למרות כלל הדברים הטובים, ישנו חסרון בולט כאשר אלו מדברים על פרויקט המכיל אלמנטים מרובים, במצב כזה ישנו עומס רב בסרגל ההתמצאות בפרויקט (דוגמת "ניהול מערכת קرونוט חכם"), בנוסף כאשר הוספנו יותר מ-2 קرونוט בארכיטקטורת ה- IBD הכללי נתקלנו בעומס רב שיצר בלגן ויזואלי שהקשה על זיהוי הקישוריות בין היחסיות השונות.

**באיור 4.17 ניתן לראות את ציון עברו המדד ה- $n$ 'ל וזואת בהתאם ליקולות שצוינו לעיל**



אייר 4.17 – ציון עבור מדריך "נוחיות ויזואלית" Enterprise Architect

#### 4.2.2 מדריך 2 - ביצוע הגדרת חוזרת / עדכון הגדרה:

ביצוע הגדרה באופן חוזר הינו תהליך פשוט יחסית, לאחר יצירת IBD לכל בלוק ב- BDD ניתן בקלות להשתמש באותו IBD פרטני ב- IBD של כלל המערכת ע"י גירתו של הישות מהתפריט הצד שמאל למערכת עצמה.

לאחר גירתו נקבל אפשרות להציג את רכיביו הפנימיים או שהוא יהיה "קופסה שחורה" לטובת נוחות.

כל פורטי הכנסה והיציאה שהגדכנו ב- IBD הפרטני של הבלוק יופיעו באלמנט שגרנוו. ניתן ליבא אלמנטים זהים ללא הגבלה כאשר לכל אחד מהם נקבל אפשרות לתת שם ייחודי לטובת זיהוי (בנוסך לשם הבלוק המקורי שהוא מקבל בירושה).

לכל אלמנט חדש שנוסף נצטרך לבצע קישור לשאר האלמנטים ב- IBD.

עדכון של בלוק (IBD) לא משפיע על ארכיטקטורת ה- IBD הכללית, שם לא נוסף אזכור לעדכון, ובמידה ורוצים שהקשר יעשה מהאלמנט המעודכן יש צורך במחיקה והגדרה מחדש

או עדכון פרטני של האלמנט ב- IBD הכללי, דבר המסביר את הפשטות שהינו מצפים לה ובעצם גורם להגדרה מחדש לישיות שכבר הוגדרו.

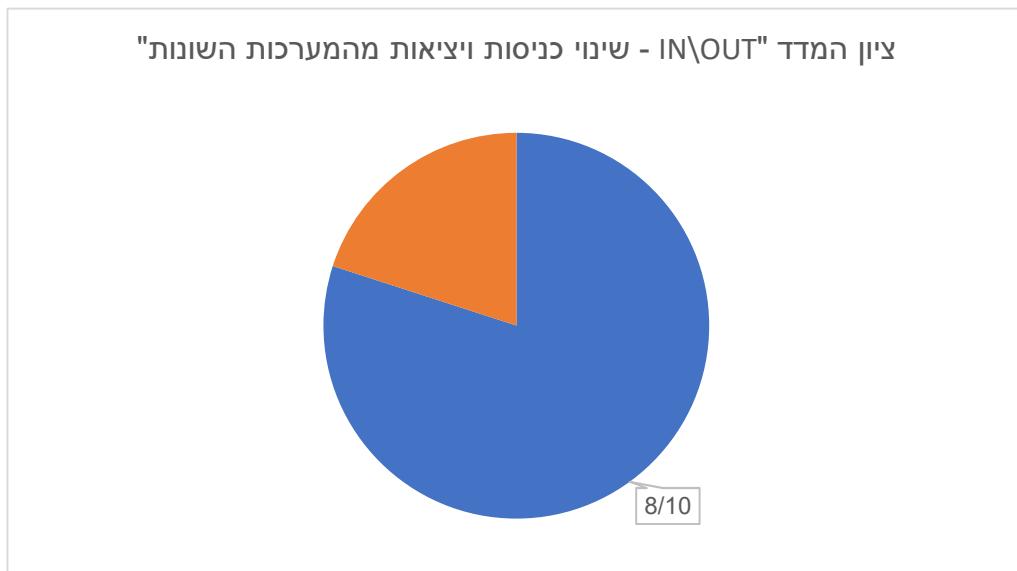
לעומת זאת ב- BDD וב- IBD הפרטני של כל בלוק ישנו קשר ישיר ושינוי באחד מהם ישפיע וישנה את الآخر.

כך שלמעשה ניתן לראות את זה באופן הבא:

1. יצירת BDD עם תיאור הבלוקים.
  2. יצירת IBD עם תיאור האלמנטים הפנימיים, הקישורים ביניהם ו קישורי ה- O/I.
  3. יצירת IBD כללי شامل אלמנטים (בלוקים מפורטים) ע"פ הזמן בו הם נלקחו.
- כאשר בין הסעיפים 1 ו-2 יש קשר ישיר, וסעיף 3 עצמאי לחלוטין.

- הציפייה הייתה יכולה גירה/ פעולה פשוטה – ידובר במדד 5 ("שינויי ארכיטקטורה באמצעות הזזה אלמנטים")

**באיור 4.18 ניתן לראות את ציון עבור המדריך הנ"ל וזאת בהתאם ליכולות שצוינו לעיל**



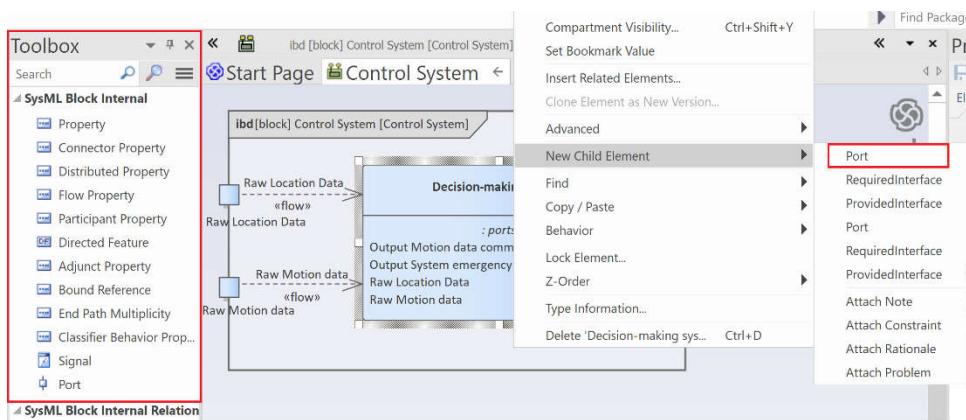
איך 4.18 – ציון עבור מדריך "ביצוע הגדרת חוותה / עדכון הגדרה" – Enterprise Architect :4.18

#### 4.2.3. מודד 3 - IN\OUT - שינוי כניסה ויציאה מהמערכות השונות:

שינוי הכניסות והיציאות במערכת הינו פשוט ביותר ביחס לאשר אלו עובדים על IBD של בלוק ספציפי ניתן למחוק/להוסף פורט O/I בקלות והמידע מתעדכן גם ב- BDD הכללי. הוספה מתבצעת ע"י גירירה של פורט מה-Toolbox או לחץ ימני ובבחירה להוספה של פורט, מחיקה מתבצעת ע"י מחקה של הפורט בתפריט (ניתן למחוק בשרטוט אך עדין הוא יהיה קיים ברכיבי הבלוק ב- IBD).

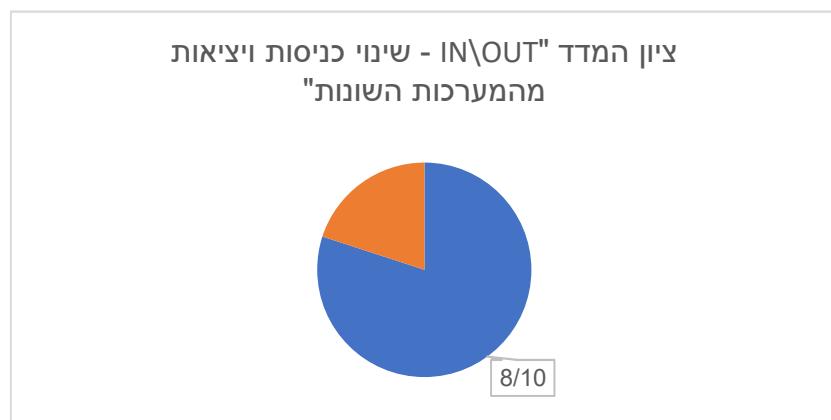
יש לציין שכאשר נוסף בלוק חדש של IBD כל הפורטים הופיעו ביחד ללא סדר של כניסה היציאות דבר אשר גורר התאמות רבות.

**איור 4.19 ניתן לראות דוגמה להוספת פורט ברכיב קיים**



**איור 4.19 - הוספת פורט ברכיב קיים - Enterprise Architect :4.19**

**איור 4.20 ניתן לראות את ציון עבור המודד הנ"ל וזאת בהתאם ליכולות שצוינו לעיל**

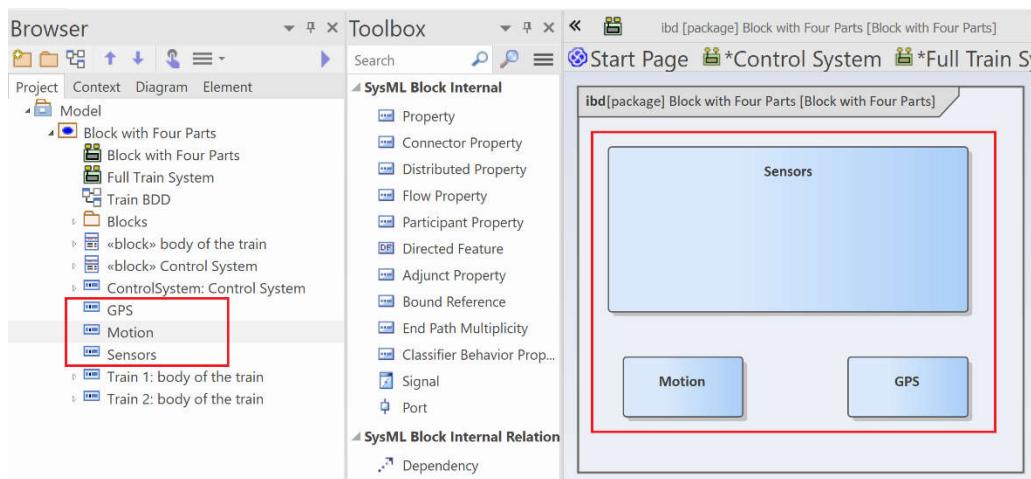


**איור 4.20 ציון עבור מודד OUT\IN - שינוי כניסה ויציאות מהמערכות השונות - Enterprise Architect :4.20**

#### 4.2.4. מדריך 4 - שינוי ארכיטקטורה באמצעות הזזה אלמנטים:

הזזה אלמנטים לטובת שינויים היא דבר לא פשוט, יש לנתק אותה מכלל הקישורים שלו, להעתיק ולהדביק אותו במקומות הרצוי, לאחר מכן יש לבצע שוב את כל הקישורים הנדרשים. כמו כן, יש להתאים כניסה ויציאה בהתאם לשינויים

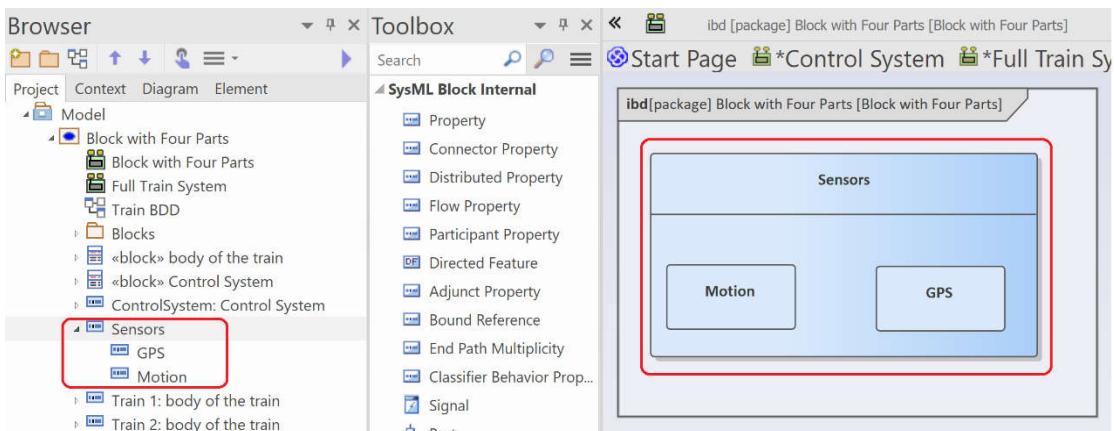
**באיור 4.21 ניתן לראות דוגמה לIBM Rational Architect עבור חישני התנועה וה-GPS, חיצוניים לבLOC החישנים**



אייר 4.21 – חישנים עצמאיים

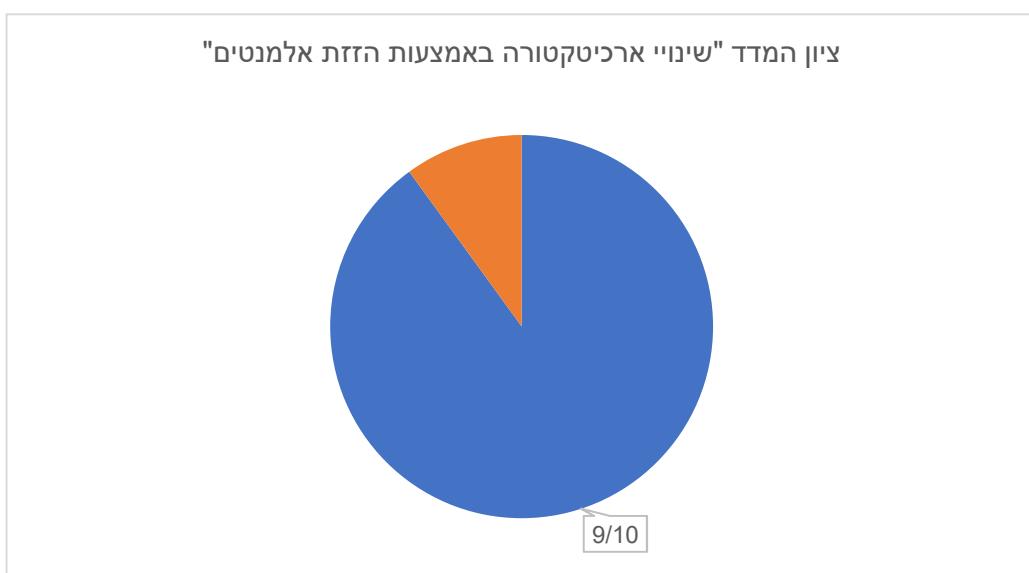
ע"י גירה פשוטה בטליה משמאלי של החישון התנועה וה-GPS בתוך בлок החישונים נוכן לראות כתם שם נמצאים בתוך בлок החישונים.

**באיור 4.22 ניתן לראות דוגמה לחישונים בתוך בлок ייעודי**



**אילו 4.22 - חישונים בתוך בлок ייעודי**

**באיור 4.23 ניתן לראות את ציון עבור המדריך וזאת בהתאם ליכולות שצוינו לעיל**



**אילו 4.23 – ציון עבור מדריך "שינוי ארכיטקטורה באמצעות הזרת אלמנטים"**

#### 4.2.5. מדריך 5 - שפת מידול מוכרת לכל המשתתפים בפרויקט

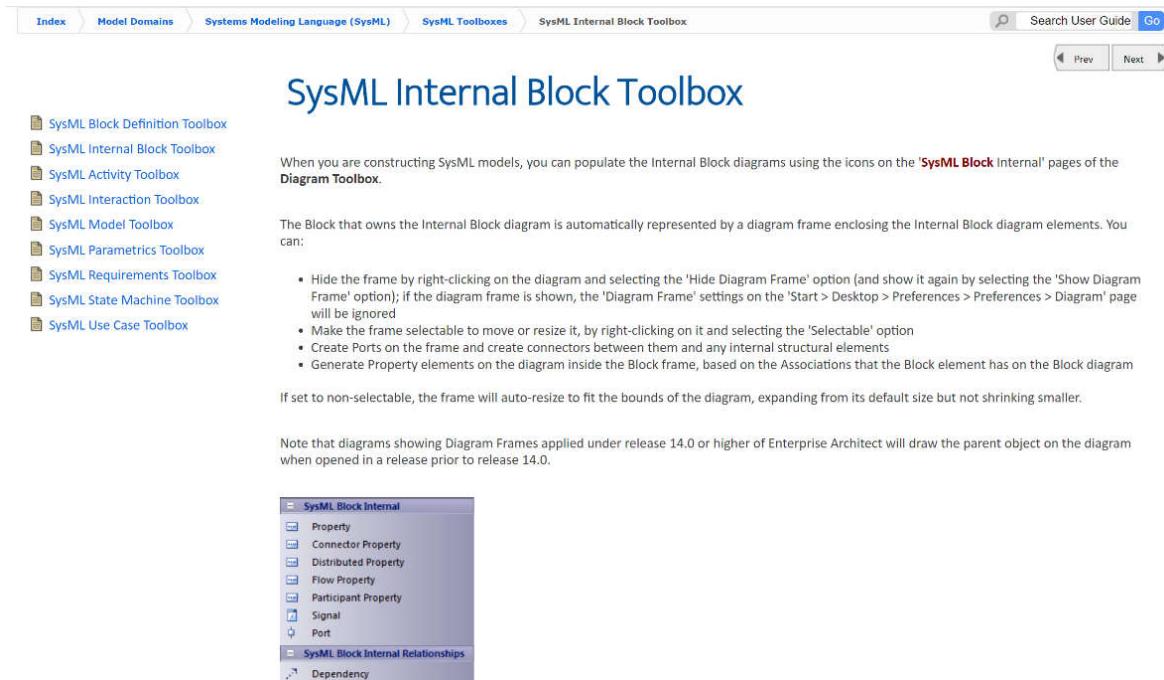
שפה המידול SysML הינה שפה מוכרת ונפוצה בקרב מהנדסי מערכות, אולם יש לה אפשרויות רבות ליישום ומידול של כל דבר. ריבוי אפשרויות זה משאיר את המשתמש עם צורך להביא מומחים שיתנו הנחיה בינה לשימוש ובמה לא. ולכן יש צורך בהבנה מעמיקה ולמידת השיטה בכדי הגיעו לניצול מיטבי של הכלים שהשיטה מציעה.

ה- (EA) Enterprise Architect מאפשר ביצוע מידול ברמה גבוהה ומוצע מגוון אפשרויות אחרות יש להכיר בכדי לבחור באפשרות הטובה ביותר.

חשוב לציין שיש ל-EA "עזרה" מפורטת וברורה ובנוספ' ישנים סרטיונים וחומר רב באינטרנט על השיטה ועל עובודה עם המערכת.

- ישנו חיסרון משמעותי מואוד ביצוע מידול ברמה הפיזית, **המידול מתבצע ברמה הולוגית בלבד**, דבר אשר מחייב התמסחות אל כלieder אחר על מנת לבצע סימולציה (התמסחות לתוכנה כמו MATLAB).

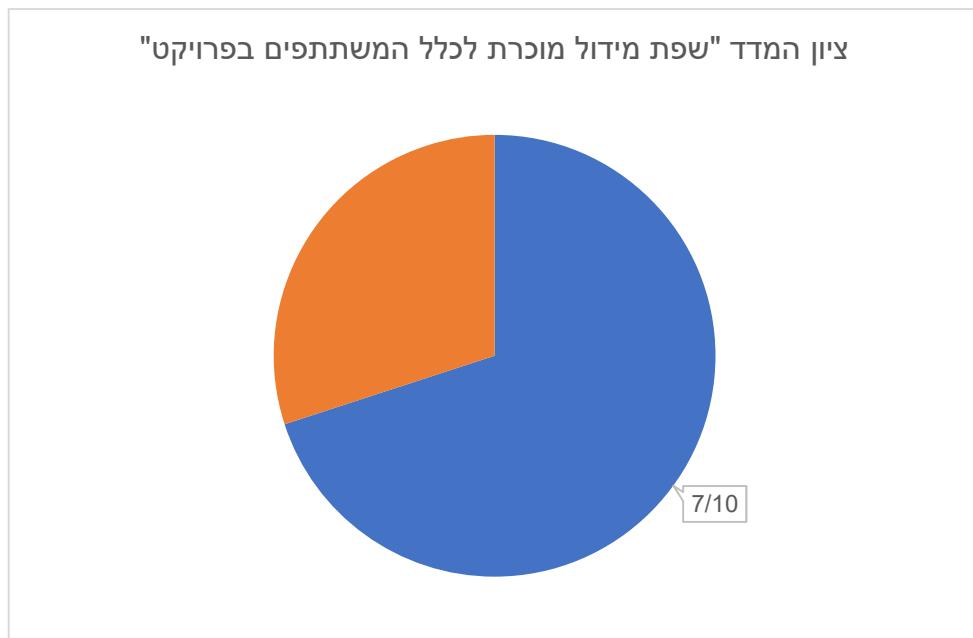
baior 4.24 ניתן לראות דוגמה למדריך המשתמש



The screenshot shows the Enterprise Architect User Guide interface. The top navigation bar includes links for Index, Model Domains, Systems Modeling Language (SysML), SysML Toolboxes, SysML Internal Block Toolbox, and search functions. Below the navigation is a breadcrumb trail: SysML Internal Block Toolbox. The main content area is titled "SysML Internal Block Toolbox". It contains a sidebar with links to other toolboxes: SysML Block Definition Toolbox, SysML Internal Block Toolbox, SysML Activity Toolbox, SysML Interaction Toolbox, SysML Model Toolbox, SysML Parametrics Toolbox, SysML Requirements Toolbox, SysML State Machine Toolbox, and SysML Use Case Toolbox. The main text area provides information about using the Internal Block toolbox to construct SysML models. It mentions that the block owning the Internal Block diagram is automatically represented by a diagram frame enclosing the Internal Block diagram elements. A list of actions for the frame is provided, such as hiding it or making it selectable. A note states that if set to non-selectable, the frame will auto-resize to fit the bounds of the diagram. A note also mentions that diagrams showing Diagram Frames applied under release 14.0 or higher of Enterprise Architect will draw the parent object on the diagram when opened in a release prior to release 14.0. At the bottom, there is a screenshot of the "SysML Block Internal" palette, which lists various internal block elements: Property, Connector Property, Distributed Property, Flow Property, Participant Property, Signal, and Port. Below this is the "SysML Block Internal Relationships" section, which includes Dependency.

baior – מדריך המשתמש :4.24 Enterprise Architect

**באיור 4.25 ניתן לראות את ציון עבור המדריך הנ"ל וזאת בהתאם לכליות שצוינו לעיל**



איפור 4.25 – ציון עבור מדריך – Enterprise Architect – ציון עבור מדריך – Enterprise Architect

#### 4.2.6. מDDR 6 - השפעה של שינויים באמצע הפרויקט:

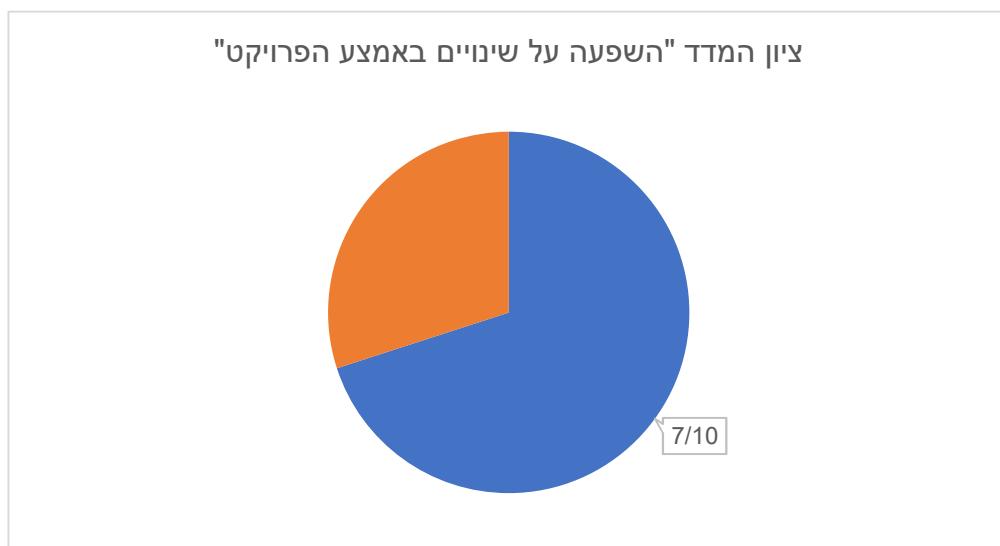
השפעה של שינויים באמצע הפרויקט היא עשויה להיות גדולה תלוי בהיקף ובסוג השינוי. כפי שדיברנו בסעיפים הקודמים, ישנה חלוקה ל- BDD כללי, ממנו נבנה IBD לכל אחד מהבלוקים ואיתם עוסוד ב- IBD הכללי כאשר שינוי ב- IBD של הבלוק לא ישפיע על ה- IBD הכללי.

לכן אם מדובר על שינוי בתוך אחד הבלוקים (לא השפעה על פורטי הכניסה והיציאה השינוי לא יורגש בארכיטקטורה הכללית, אולם אם השינוי ישפיע על פורטי הכניסה והיציאה ההשפעה תורגש ותדרוש מעתנו לבצע את השינוי ב- IBD הכללי באמצעות הצורות הבאות:

1. מחיקה של האלמנט והוספתו מחדש (דורש מחיקה של קישוריםים והגדלתם מחדש).
2. הוספה ידנית של השינוי וטיפול בקשריות לשינוי בלבד.

שיטת 2 היא עדיפה, אך בפרויקט מסווג שבו יש מערכת מרכזית עם סיבוכיות גבוהה בעלת הרבה תתי מערכות עם סיבוכיות נמוכה שינוי מסווג זה עשוי להיות גדול ולדרש עבודה רבה.

#### באיור 4.26 ניתן לראות את ציון עבור המדריך וזאת בהתאם ליכולות שצינו לעיל



איך – ציון עבור מדריך "השפעה על שינויים באמצע הפרויקט" :4.26 Enterprise Architect

## 4.2.7 יתרונות ל-Enterprise Architect

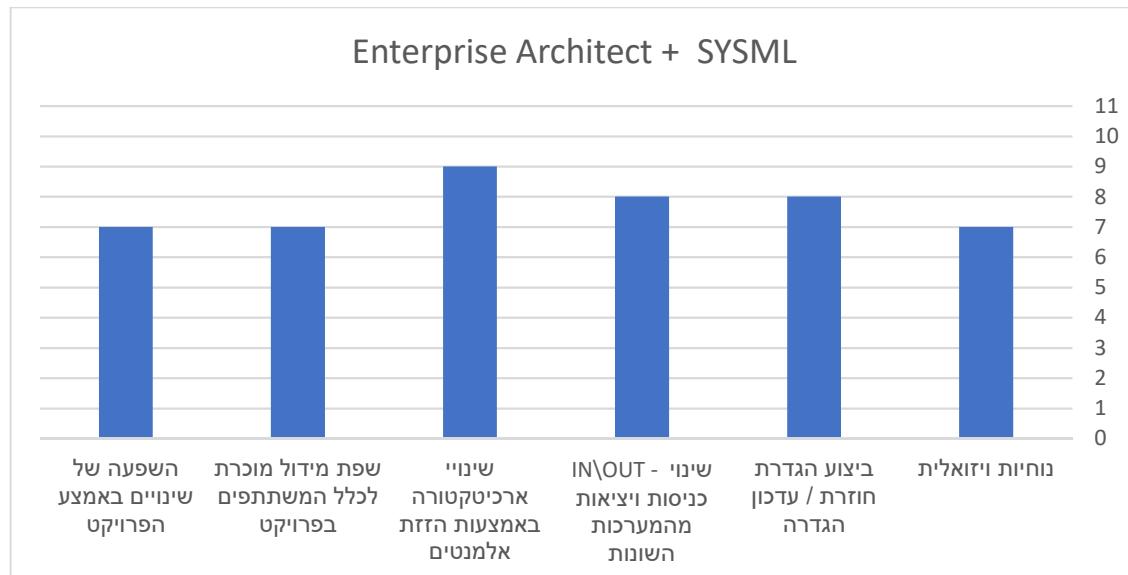
- אלמנט שחזור על עצמו נדרש במבנה פעם אחת בלבד.
  - קישור פשוט בין האלמנטים השונים.
  - הוספה והסרה של רכיבים בצורה פשוטה ומהירה.
  - עובדת בצורה היררכית מסודרת (בהתאם לשפת SysML).
  - מעבר בצורה פשוטה בין הרמות השונות.
  - תצוגה ויזואלית ברורה.

4.2.8 חסימות ל-Enterprise Architect

- שינויים גדולים בזמן לאחר המידול עשויים להיות מרכיבים.
  - שינויים ב-IBD של בלוק לא משפיע על ה-IBD הכללי.

- יש לציין שההשלטה באיזה סדר לבנות את המודל ובאיזה חלקים של SYML שתשמש נشارת בידי מהנדס המערכת. נקודה זו יכולה להיחשב כगמישות אך למעשה יכולה גם להפריע ולבבל את מהנדס המערכת (חומר היכרות את כלל האפשרויות וכו').

4.27 גראף המתאר את ציון המדדים ניתן לראות באIOR



איך Enterprise Architect + SysML :4.27 - גוף ציוני מדדים

**7.5** שקלול הציון עבור הכלי Enterprise Architect בהתאם לחלוקת המדים הוא:

- בסעיף זה נציג את כלי ה-Visio, שבשונה מהכלים האחרים אינו כלי מידול אלא תוכנת שרטוט ולכן ההגדרות הן ברמה מופשטת יותר (שמות ואלמנטים) ובין זהות להגדרות אחרות ניתן לבצע בתוכנות המידול אחרות שנבדקו.

#### 4.3.1. מדריך 1 - נוחיות ויזואלית:

מבחן ויזואלית, ממשך המשמש נכון באופן יחסית, כל עוד השרטוט פשוט אז המראה ברור וקריא, ככל שההרטוט מסתובב ונעה מרכיב, כך הצורה היזואלית נהייה קשה יותר, בנוסף אם זה חלק ש חוזר על עצמו מספר פעמים ניתן פשוט ליצור אותו פעם אחת ולכתוב number-X על מנת ליציג כמה פעמים הוא צריך על עצמו (כמו פרויקט הקرونוט).

כמו כן, קיימים שם מייצגים לכל "רכיב", כאשר כלל מרכיבי המערכת מקושרים ביניהם באמצעות קישור אשר עליו יש את שם הזרימה והוא על מנת להזמין את הקישוריות בין 2 מרכיבים, בנוסף ישנה אפשרות לחזוץ על הקישור, אשר בעת הלחיצה מסומן בצבע בולט.

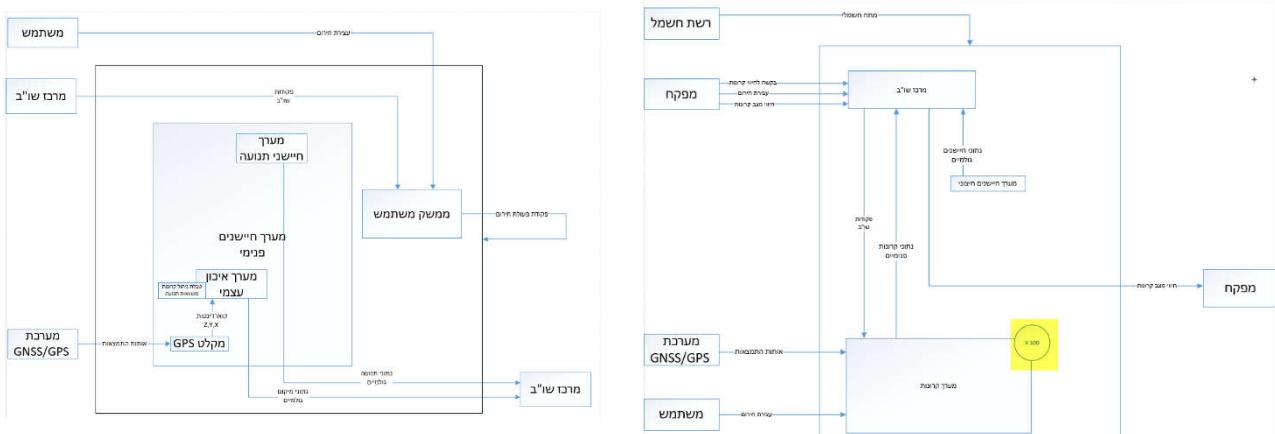
אין אפשרות לצלול פנימה, כאשר רוצים לשרטוט רכיב בתוך רכיב זה נעשה בצורה מסובכת אשר מקטינה את כלל השרטוט והופכת אותו ללא קריא. בנוסף, המעבר בין מערכות שונות הוא באמצעות TABs דבר אשר יוצר חוסר היררכיה.  
אין היררכיה כלל בשרטוט למעט רכיב בתוך רכיב.

למרות החשיבה המקובלת אצל אנשים רבים שעובדים עם Visio שיתרונותיו הם פשוטותו.

דוגמת פרויקט "ניהול מערך קرونוט חכם", כאשר ניתן לשרטוט את הקרון פעם אחת כתת מערכת ולציין אותו X (כפול) מספר פעמים לארכיטקטורת העל. הניסוי המעשני והניתוח שביצענו מראה שכאשר רוצים להגיע לפירוט ברמה גבוהה של מערכות במורכבות גבוהה או בינונית, יתורן זה אינו קיים, כמו בדוגמה הקרון שפורטה לעיל, במקרה שלנו, רק כאשר רצינו להכפיל את הקרון שיצרנו בכל המידול SC, גילנו שהכניות והיציאות אינן מוגדרות כראוי ושישנו הבדל גדול בין תכנון של קרון בודד X (כפול) מספר הכפלות לבין 2 קرونוט ומעלה X (כפול) מספר הכפלות.

عقب השימוש בכל שרטוט לא יכולנו לראות את אופן שליחת הנתונים מהקرونוט (וקטוריים) אל מרכז הבקרה, לא הי ניתן להוציא דוח מפורט על כל קרון/קישור, וכך כן, ככל שישנם היררכיות רבות יותר כך השרטוט יהיה קשה יותר להבנה, חסכנות שועלמו מיד ברגע המעבר אל כל המידול וביצוע סימולציה ראשונית לצורך בדיקת היתכנות.

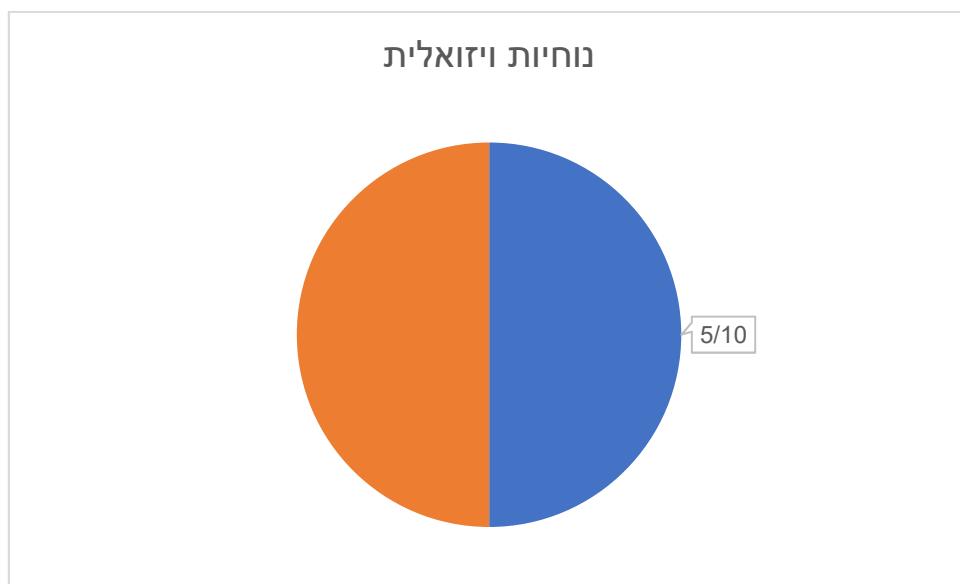
באיור 4.28 ניתן לראות דוגמה לארכיטקטורה כללית אשר בתוכה מערך קורנות X100 קורנות (בצהוב) ובאיור 4.29 ניתן לראות את מערך הקורנות עצמו, אשר שורטט פעם אחת.



איור 4.29 – מערך קורנות (קנון בודד)

איור 4.28 – ארכיטקטורה כללית של ניהול מערך קורנות חכם

באיור 4.30 ניתן לראות את ציון עבור המدى הנ"ל וזאת בהתאם ליכולות שצוינו לעיל



איור 4.30 – ציון עבור ממד "נוחיות ויזואלית"

### **מדד 2 - ביאוש הגדרת חנורת / עדכנו הגדרה: .4.3.2**

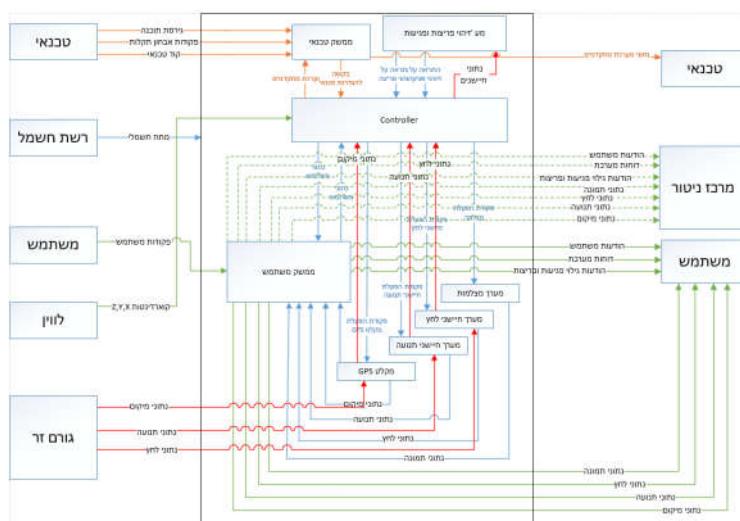
ביצוע הגדרות באופן חזרה היה די פשוט, הבעייה הייתה הינה סרבל השרטוט עקב כמוות הרכיבים באותו השרטוט.

כמו כן, אין הגדרות שניתן להפוך למשמעות, דבר אשר יוצר פעללה חוזרת עבור כל רכיב ומאריד את התהילה.

כמו כן, שינוי ארכיטקטורה באמצעות הפרויקט הוא דבר לא מסובך כל כך, שינוי זה מצריך שנייתן לייצר לעומת כלים אחרים בנוסח, עדכון הגדרה גם הוא היה פשוט למדי וזאת בגלל חוסר ההגדרות והאפשרויות

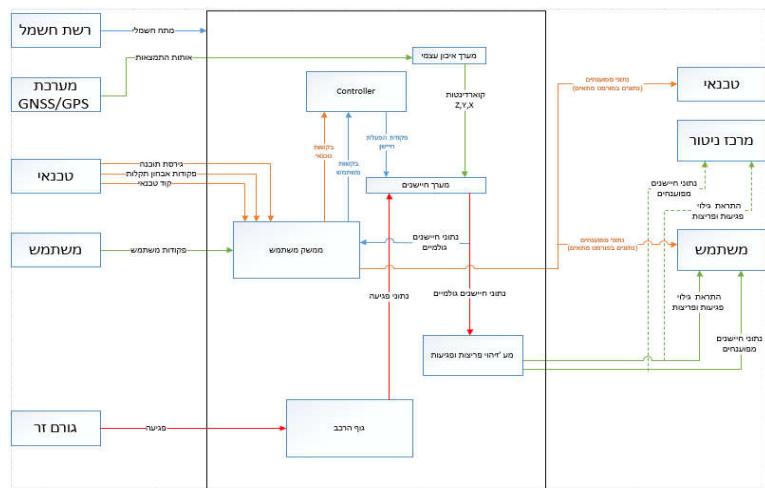
אולם כפי שאמרנו בסעיף הקודם, למטרות פשוטתו של הכליל במערכות בעלות מורכבות גבואה או בינונית יתרון זה אינו קיים

באיור 4.31 ניתן לראות דוגמה למצב היישן של הארכיטקטורה הכללית



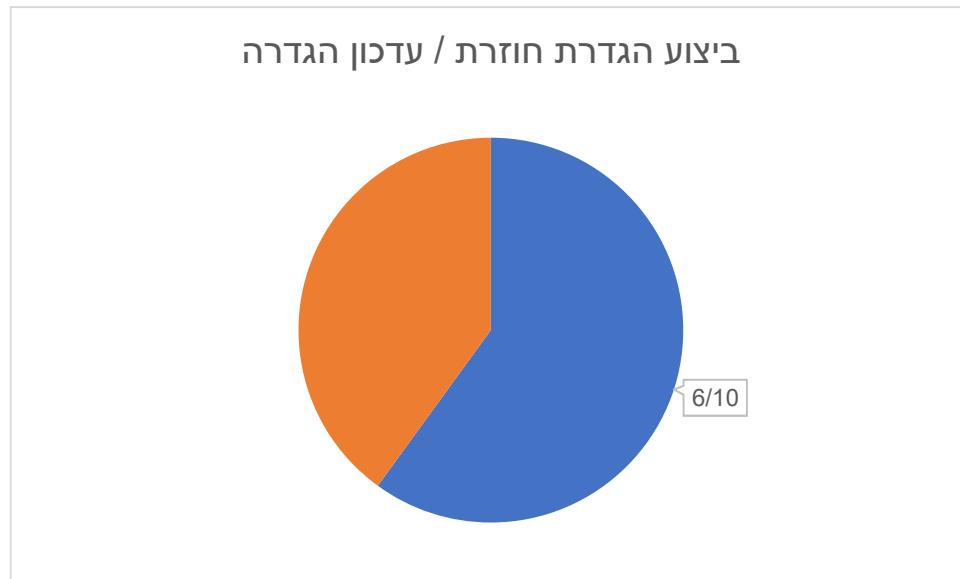
איור 4.31: Visio - ארכיטקטורה מבכ' ישן

**באיור 4.32 ניתן לראות דוגמה למצב החדש של הארכיטקטורה הכללית**



איור 4.32 Visio - ארכיטקטורה מצב חדש

באיור 4.33 ניתן לראות את ציון עבור המדרן הנ"ל וזאת בהתאם ליכולות שצוינו לעיל



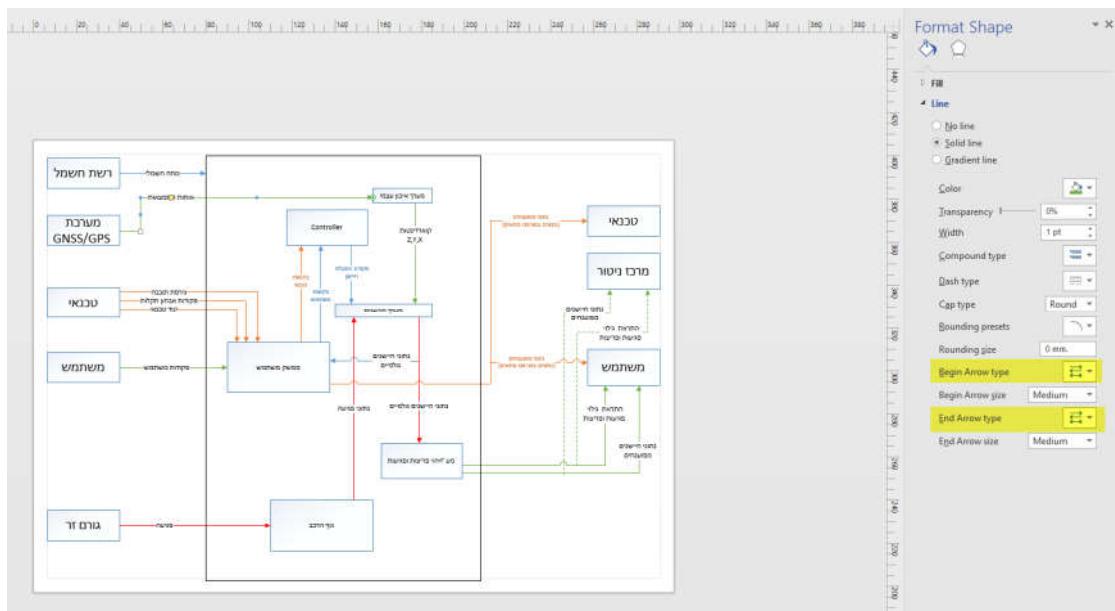
**איור Visio :4.33 – ציון עבור מדריך ביצוע הגדרת חוזרת / עדכון הגדה**

#### 4.3.3 מדד 3 - **IN\OUT** - שינוי כניסה ויציאה מהמערכות השונות:

שינויי הכניסות והיציאות במערכת הוא נושא פשוט מאד, הדבר מצריך רק שינוי של כיוון החץ (כניסה או יציאה), בעצם באמצעות שינוי החץ ניתן לשנות את סוג הכניסות והיציאות.

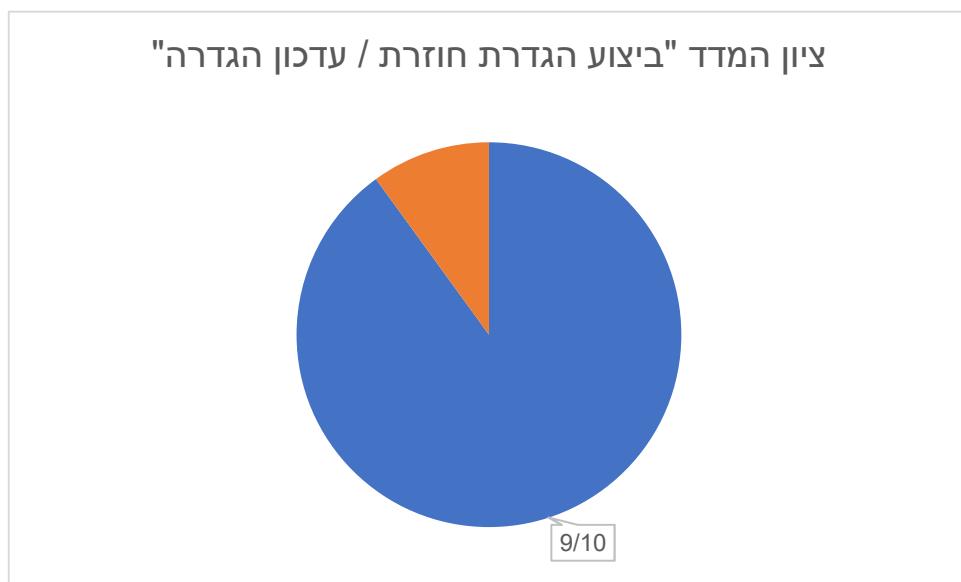
אך עם זאת, אין יכולת הגדרה כללית, אין יכולת הגדרת הטווח ומה עבר על הפורט (אפשר לכתוב במלל על כל פורט בנפרד), דבר אשר מסרב מאוד את השירות ולוקח הרבה זמן

**אייר 4.34** ניתן לראות דוגמאות לשינוי כיווני הכניסות והיציאות



אייר 4.34 - שינוי כיווני חצים

**באיור 4.35 ניתן לראות את ציון עבור המדריך הנ"ל וזאת בהתאם לכליות שצוינו לעיל**



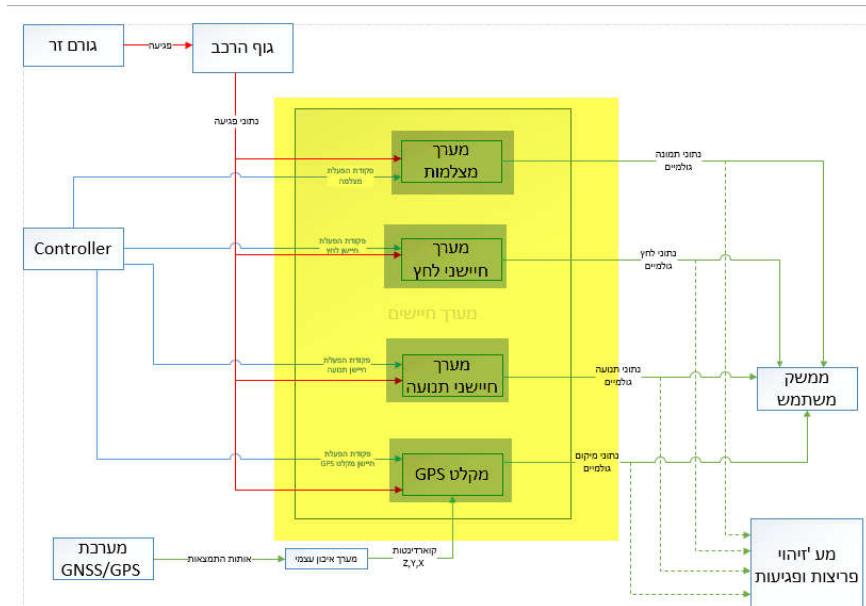
איפור 4.35 – ציון עבור מדריך "ביצוע המדריך חוזרת / עדכון הגדרה" – Visio : 4.35

#### 4.3.4. מדריך 4 - שינוי ארכיטקטורה באמצעות הזרת אלמנטים:

פעולה זו מוגבלת מאוד, שכן אין אפשרות ליצור היררכיות איזי הכל מוצג ברמת על/צלילה לכל רכיב ב-TAB אחר, ההזזה הפיזית של האלמנטים נעשתה בצורה רגילה כמו בכל kali, لكن הפעולה עצמה פשוטה מאוד, אך מלכתחילה אינה עונה על הדרישה של ההיררכיה.

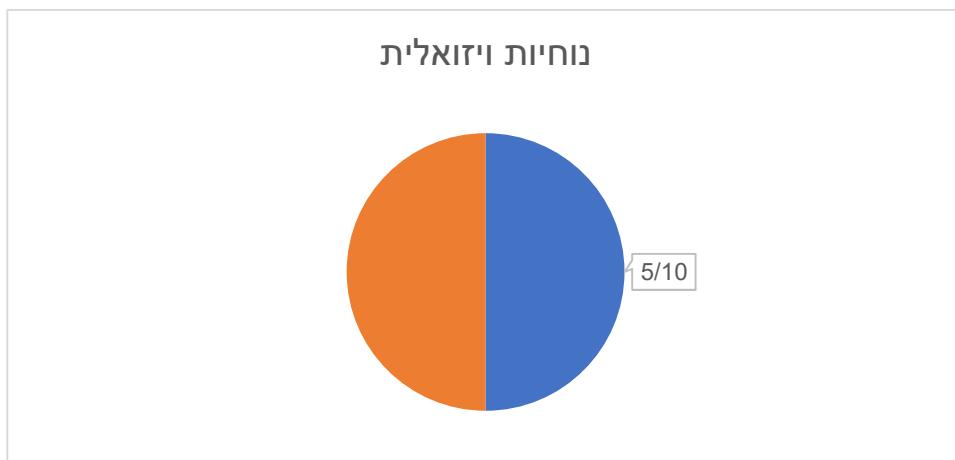
כמו כן, יש להתאים כניסה ויציאה בהתאם לשינויים.

באיור 4.36 ניתן לראות דוגמה לארכיטקטורה הכוללת



איור 4.36 - ארכיטקטורה כללית Visio :4.36

**באיור 4.37 ניתן לראות את ציון עבור המדריך נ"ל וזאת בהתאם לכליות שצוינו לעיל**



אילר 4.37 – ציון עבור מדריך "ציון המדריך" שנויי ארכיטקטורה באמצעות היזת אלמנטי Visio

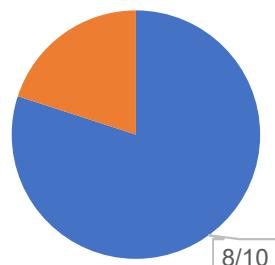
#### 4.3.5. מדריך 5 - שפת מידול מוכרת לכל המשתתפים בפרויקט:

בנייה האררכיטקטורה הינה פשוטה יחסית.

שפה המדול מובנת ופשוטה. במקרה שהיינו רוצים לתאר רק מבנה אז"י היינו יכולים אולי לחת ציון 10. אך מכיוון שישנו צורך גם בהציגת ה"לוגיקה" יש לצוין שקיימות תבניות שאפשרות זאת, בלי אפשרות סימולציה כMOVEN. אבל לצורך קידור להכין שרטוטים נפרדים וזאת עקב העובדה שהיררכיה שצוינה במדד מס' 1 לעיל. כמו כן, ניתן להציג חלקיים של המודול בשרטוטים נפרדים ולהסביר את היררכיה במלל אבל אם יתבצעו שינויים התחזקה תהיה מסורבלת ביותר. למדד זה לא נלקח בחשבון כאמור נושא יכולת סימולציה מכיוון שאינה קיימת כלל בכלל

**באיור 4.38 ניתן לראות את ציון עבור המדריך ה"ן" וחת צוין בהתאם ליכולות שצוינו לעיל**

ציון המדריך "OUT\IN" – שינוי כניסה ויציאה מהמערכות השונות"



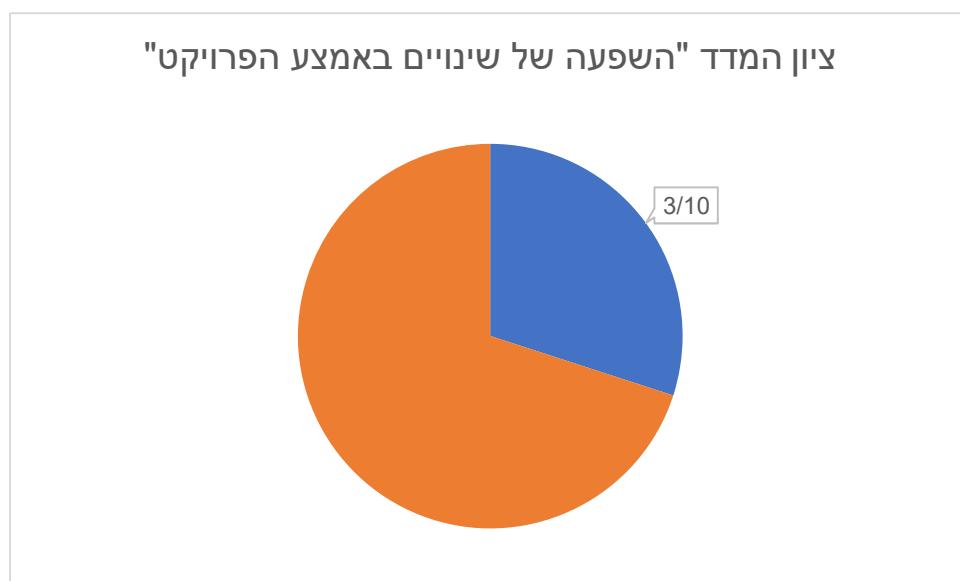
איור 4.38 – ציון עבור מדריך "OUT\IN" – שינוי כניסה ויציאה מהמערכות השונות" Visio : 4.38

**4.3.6. מDDR 6 - השפעה של שינויים באמצע הפרויקט:**

השפעה של שינויים באמצע הפרויקט היא גדולה ורחבה היקף, לאחר שבוצעו הרבה הגדרות, קישוריהם, TABs וכי', הדבר ככל הנראה יצריך שינוי מסיבי של כל האלמנטים בכל המומות! האלמנטים שככל הנראה נצטרך לשנות ולעורך הם: כניסה ויציאה, קישורים, הזנת רכיבים, ולבסוף זאת בכל TAB באופן נפרד. השינוי הדרוש המונע עבודה בכל רכיב המערכת, שינוי זה ארוך מאוד ולא פשוט, יש מקום רב לטעויות אנוש.

○ ראה תМОונות בסעיף 3

**באיור 4.39 ניתן לראות את ציון עבור המDDR הנ"ל וזאת בהתאם ליכולות שצוינו לעיל**



איור 4.39 – ציון עבור מDDR "השפעה של שינויים באמצע הפרויקט" Visio : ציון עבור מDDR "השפעה של שינויים באמצע הפרויקט"

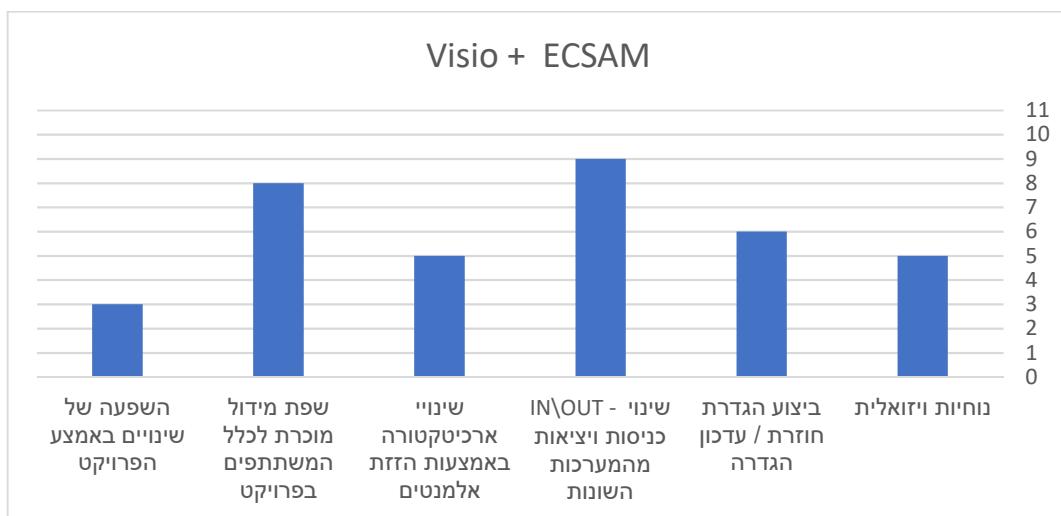
#### 4.3.7. יתרונות ל- Visio :

- המוביילה בשוק בתוכנות דיאגרמות
- בעל מגוון רחב של צורות דיאגרמות
- רוב אנשי המקצוע כבר מכירים את התוכנה וכי怎ד להשתמש בה
- פועל רק בסביבת עבודה של Windows
- הוספה והסרה של רכיבים בצורה פשוטה ומהירה
- קישור בצורה פשוטה בין הרכיבים השונים
- שכפול אלמנטים גדולים ומורכבים (פרויקט ניהול מערכ קיוניות חכם),
- יכולת שיתוף הפרויקט באופן פשוט ללא צורך בהורדת/קיית התוכנה
- ממשק WEB נוח עבור שיתוף ארכיטקטורת הפרויקט לשימוש ללא Visio

#### 4.3.8. חסרונות ל- Visio :

- בניית המודל לוקח זמן רב על מנת לעצב אותו לרמה קריאה וברורה
- יכול להשתנות מספר פעמים, עקב בניה לא קריאה
- שינוי ארכיטקטורה עלול להיות מסובך מאוד
- אין יכולת קבלת נתונים על כל קישור (פרמטרים, פרוטוקול, וכו')
- יש לכתוב על כל קישור את הנתונים באופן סטטי-קבוע
- אין אפשרות לחבר מסמכים לדיאגרמות השונות במערכת
- אין אפשרות ביצוע סימולציות וכותזאה מכך אין יכולות ניתוח

graf המתאר את ציון המדרדים ניתן לראות באיוור 4.40



איוור 4.40 - גף ציוני מדדים

**شكلול הציון עבור הכללי Visio בהתאם לחלוקת המדדים הוא: 5.3**

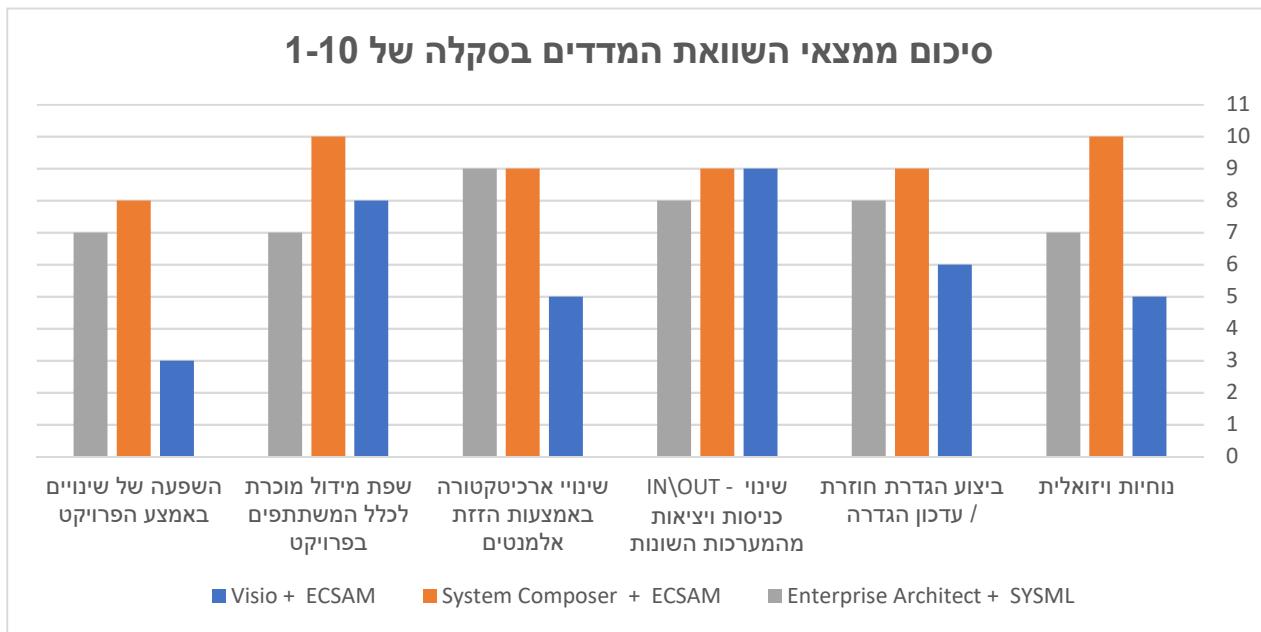
#### 4.4. סיכום ממצאי השוואת המדדים בסקלה של 1-10 (1=לא טוב, 10=מצוין)

להלן טבלה 4.2 המשכמת את הציון עבור כל מדד עבורי כל כלי בנפרד

Enterprise Architect + SysML	System Composer + ECSAM	Visio + ECSAM		מדד מס' מדד
7	10	5	נוחיות ויזואלית	1
8	9	6	ביצוע הגדרת חוזרת / עדכון הגדרה	2
8	9	9	IN\OUT - שינוי כניסה ויציאה מהמערכות השונות	3
9	9	5	שינויי ארכיטקטורה באמצעות הזזה אלמנטים	4
7	10	8	שפה מידול מוכרת לכל המשתתפים בפרויקט	5
7	8	3	השפעה של שינויים באמצעות הפרויקט	6

טבלה 4.2: טבלת סיכום ציוני מדדים

**גרף סיכום המתאר את ציון המדרדים המשוכלל של כל הכלים שנבדקו ניתן לראות באIOR 4.41**



להלן טבלה 4.3 המסכמת את תוצאות המדרדים המשוקללים עבור כל כלי + שיטה בנפרד:

תוצאה סיכום מדרדים משוקל	כלי ושיטה
9	System Composer + ECSAM
7.5	Enterprise Architect + SysML
5.3	Visio + ECSAM

טבלה 4.3: טבלת סיכום ציונים משוקללים

#### 4.5. יכולת אימות הניתוח על ידי סימולציה

מפתחת גודל המחקר, הזמן והמשאים המוגבלים סעיף זה נבחן רק ברכי המידול העיקרי אשר נחקר בפרויקט, ה- System Composer. כמו כן, נושא סימולציה עם כלים תומכי SysML נבחן בעבודות אחרות (ראו פרק 2.3).

במערכת ישנן אפשרויות רבות לטובת יצירת תרחישים ונקודות דוגמה במהלך הסימולציה, ביצוע סימולציות פשוטות ניתנות לביצוע בצורה קלה יחסית כאשר סימולציות מורכבות יותר דורשות הבנה והכרה של המערכת ויכולותיה.

קיבלת התוצאות מהסימולציה הינה פשוטה, ישנה אפשרות לראות כל שלב ושלב באופן איטי, להקליט, ולנתח תוך כדי ההרצה עצמה.

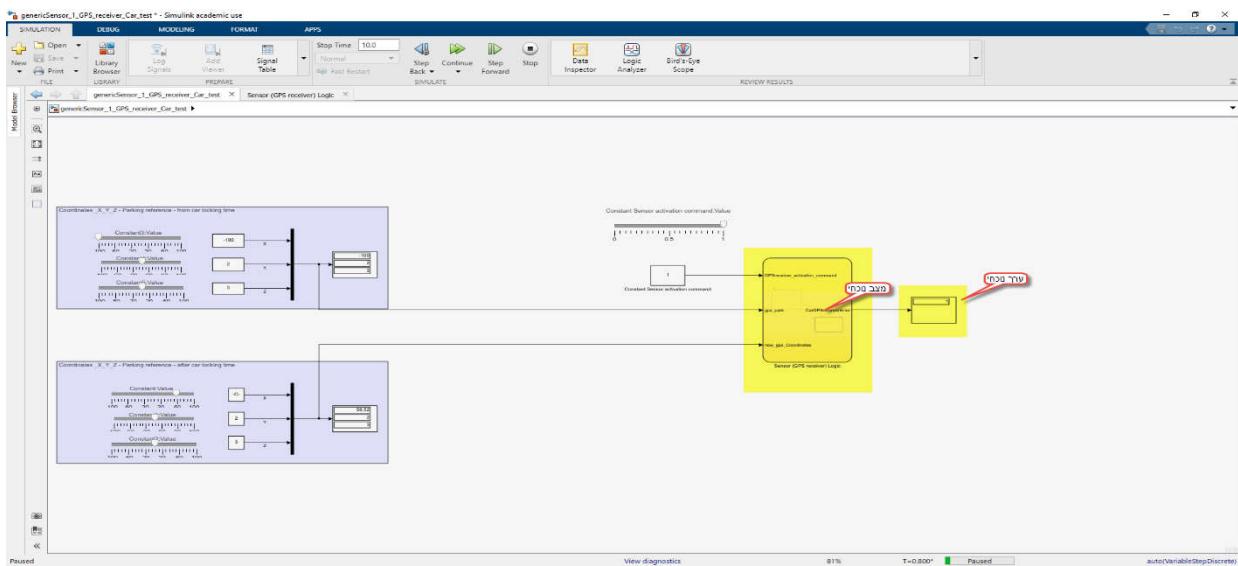
כמו כן, בזמן ריצת הסימולציה ניתן לראות את המעברים בין המצבים השונים, את הערכים אשר משתנים ממצב לעוד רבות, דבר אשר מייצר תמונה רחבת וברורה הרבה יותר מאשר עבור כל המעורבים.

בנוסף, כפי שביצעו בפרויקט הזה, מכיוון שאין לנו צורך למדל כל פרט במערכת לצורך ביצוע המחקר, רישום מצאים והסקת מסקנות (כלומר, לא נעשה מידול מלא של הפיזיקה עבור דוגמאות ה- רכב וה- קרונות), ובעצם אין לנו מאיפה לקבל את האותות האמתיים והערכים בהדמייה של מימוש מלא של המערכת בזמן אמיתי, ישנה דרך לפשר את הנושא ובעצם לייצר את האותות ולדמota את הערכים, לשנות אותן תוך כדי הסימולציה ובעצם לבדוק את כלל המצבים והתרחישים השונים, זהו יתרון משמעותי של הכלים System Composer והכלים המשולבים בו.

ניתן לבצע זאת באופן ידני או באופן אוטומטי, במקרה שלנו על ידי הכנת התרחישים מראש בקובץ CSV אשר יטען אל ה- System Composer ובעצם ירוז באופן אוטומטי.

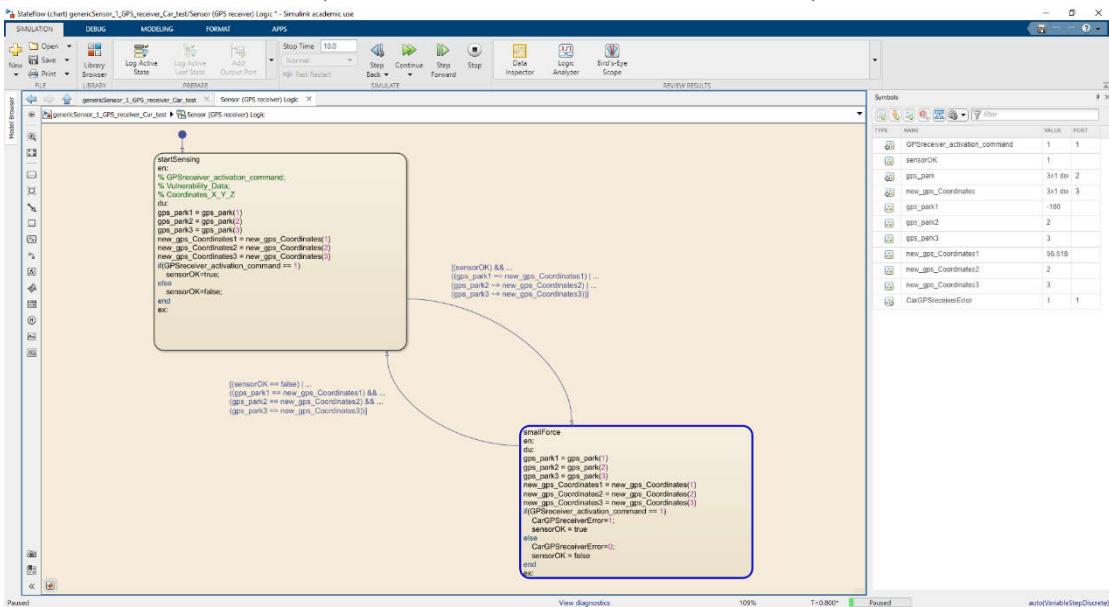
אחד היתרון של ביצוע סימולציה באמצעות כלים המידול הוא יכולת לבצע סימולציה על כל רכיב או תת רכיב בנפרד בנוסף לSIMULIA היכולת של כל הרכיבים ביחד.

### באיור 4.42 ניתן לראות דוגמה להכנסת ערכים בסימולציה



אייר 4.42 – סימולציה – הכנסת ערכים – System Composer :4.42

באירור 4.43 ניתן לראות דוגמה ל state flow ותוצאות בזמן אמת



איירור 4.43 - סימולציה - System Composer : State Flow ותוצאות

בסעיף זה נתנו מענה לשאלות המחקר המשניות 4 ו- 5

שאלת מחקר משנית 4:

- כאשר מביצעים סימולציה, באיזה שכבת מערכת יש לבצע את הסימולציה (באיזה רמה יש לבצע את התרחישים) ?

התשובה לשאלת המחקר המשנית מס' 4 כМОבן תלויה בפרויקט עצמו,

בגודלו, בהיקפו, במורכבותו ובعود פרמטרים רבים.

בהתבסס על הדוגמאות אשר השתמשנו בפרויקט הנ"ל, הרזולוציות בהן ירידו פרטים וביצעו את הסימולציות והתרחישים היו בעצם עד שלב הרכבים עצםם.

בהנחה שלא אנו ייצרנו את החישנים ו/או הרכיבים השונים בפרויקט, לא היה לנו צורך לבדוק אותם, השתמשנו בהם כמערכת מוכנה (ע"פ הגדרות היצרן), וכן בנינו את המודל עד הרכיבים הקיימים אשר נבדקו בנפרד במקום הייצור והפיתוח שלהם.

כמו כן, רכיבים אלה שולבו לאחר מכן בסימולציה ובתרחישים הכלליים, כיוון שהוא חשוב מאוד לדעת כיצד הם משפיעים על המערכת אותה אנו מפתחים.

המחקר שלנו מראה שכלים שמאפשרים זאת באמצעות שיטה מוגדרת ונוחות יחסית, במקרה שלנו: System Composer and ECSAM מאפשר לביצוע מידול של כל הרכיבים כדי לקבל רמת סמך טובה של סימולציה מערכתיות תוך שימוש בתנוני אמת (או בתנוני השלהמה למשל של פיזיקת רכיבים שאין צורך למדל כמפורט לעיל)

שאלת מחקר משנית 5:

- עד איזה שכבה יש ערך מוסף לביצוע הסימולציה ?

כמו בשאלת המחקר המשנית מספר 4, גם כאן בתשובה לשאלת המחקר המשנית מספר 5 תלואה כМОבן בפרויקט עצמו, בגודלו, בהיקפו, במורכבותו ובעוד פרמטרים רבים.

באם נתיחס אל הדוגמאות בהן השתמשנו בפרויקט הנ"ל ניתן לראות שירidea לפרטים עזורה לנו מאוד, ככל שירידנו יותר ויוטר ברזרולציות של הסימולציה, כך מצאנו יותר בעיות, אם זה בארכיטקטורה עצמה, בתרחישים ועוד.

מצאים אלה (במקרה של פרויקט אמיתי שהולך לפיתוח) היו חוסכים המון כספ, זמן, משאבים, ובעיקר שינויים באמצע הפרויקט, שינויים שלפעמים יכולים "לקבור את הפרויקט", הממצאים שנמצאו עוד לפני תחילת פיתוח הפרויקט שוויים הרבה מאוד כספ וזמן ונוטנים לנו את יכולת עלות על בעיות קריטיות עוד לפני שהגענו אליו ובמידה רבה אף לפטור אותן.

כלי הסימולציה System Composer, בעצם מעניק לנו יכולת ביצוע סימולציות מורכבות מאוד, ניתוח קל ופשוט שלהן, תוך בניית המודל והרצה שלו בזמן יחסית מהיר מאוד.

המצאים מהניסויים המיעדים שביצענו עם המודלים בכלים אפשרים להסיק לגבי נושא הסימולציה הינם:

ישנה חשיבות רבה לבחינה וביצוע סימולציה נפרדת לרכיבים השונים אשר הקלטים והפלטים שלהם משפיעים על קלטי ופלטי המערכת (דוגמהו החישונים אצלנו שימושיים על קבלת החלטה ופעולת המערכת).

לאחר בוחנה של כל אחד מהחישונים ובחינה של נקודת ההחלטה העיקרית במערכת, ע"י סימולציות פרטניות, שילבנו את כל הרכיבים וביצענו סימולציה כללית.

בעזרת הסימולציות הפרטניות זיהינו בעיות ושגיאות, בנסיבות יחסית, אותן ניתן צריכים לתקן.

כאשר אם הינו מתחילה מסימולציה כללית יהיה לנו קושי בזיהוי ואיתור הבעיות והדבר היה גוזל זמן יקר.

יתרונן נוסף בccoli ובפירוק לSIMOLIZATIONS פרטניות לפני מעבר לSIMOLIZATIONS כוללת הינו יכולת לזהות נקודות קריטיות אותן יש לבדוק והוספקן לתרחישים הכללים שנבחנים בסימולציה מערכתי הcomplete, הוספת תרחישים אלו עזרה לנו לבצע SIMOLIZATIONS כללית מיטבית יותר אשר מדמה מגוון רחב יותר של מצבים אפשריים.

בטבלה 4.4 להלן נוכל לראות סיכום של שלב הסימולציה ע"פ פרויקט והערך המוסף של הסימולציה:

סוג	שם פרויקט	רמת סימולציה	ערך מוסף
מערכת מרכזית עם סיבוכיות ביןונית/גבוהה בעלת הרבה תת-מערכות עם סיבוכיות ביןונית/גבוהה	NEYTOR פגיעה ברכב 360	סימולציה עד רמת התת רכיב - חישון בודד	יכולת זיהוי מוקדם של התנהוגות כל חישון ע"פ קלטיו בהתאם לספרים אשר נקבעו לו
		סימולציה של רכיב קבלת החלטות	יכולת לנתח ולטיעב את קבלת החלטות לפני קבלת נתוני אמת
		סימולציה כוללת	יכולת לראות את השילוב בין כל רכיבי המערכת והתנהוגות בתרחיש למול המזופה
מערכת מרכזית עם סיבוכיות גבוהה בעלת הרבה תת-מערכות עם סיבוכיות נמוכה	ניהול מערכ קرونנות חכם	סימולציה עד רמת התת רכיב לcronon בודד	יכולת זיהוי התנהוגותcronon בודד וההתאמתו לדרישות ולצפויות בשלב מוקדם
		סימולציה של רכיב קבלת החלטות	יכולת לנתח ולטיעב את קבלת החלטות לפני קבלת נתוני אמת
		סימולציה כוללת עם כל מערכ הcrononoot	יכולת לראות את השילוב בין כל רכיבי המערכת והתנהוגות בתרחיש למול המזופה

טבלה 4.4: טבלת סיכום סימולציות לפי פרויקט

## 5. פרק 5 – סיכום

### 5.1. מסקנות עיקריות מהפרויקט

1. ישנה עדיפות רבה שלב השירותים ה"חופשיים" יבנה בשיקול דעת וכובד ראש רב אך עם זאת, לא לאפשר לחלק זהה לגוזן זמן יקר מהפרויקט, שכן השירותים הללו ישתנו במספר פעמים, ואם יבנו בכל מידול אשר כולל את הסימולציה כמו ה- System Composer השירותים יבנו מחדש בכל מקרה, ככל ש实施细则 הסימולציה ישפרו את הבנת הנדרש במודל ובמערכת עצמה.

- יש לציין שסעיף זה מתייחס לפרויקטים קטנים וגדולים כאחד ניתן לראות את דוגמת הקرونוט, גם אם היינו צריכים ליצור רק קרון אחד (יחיד), ללא מערכת הבקרה, עדין היה כדאי לבצע את השירותים המעניינים (לאחר השירותים הראשוניים) בכל מידול וזאת מכיוון שהוא נוח יותר ויזואלית, בעל יותר פרמטרים אפשריות, בעל יכולות הפקט דוחות ועוד.

2. עבודה עם כל מידול בודד מתחילה הפרויקט ועד שלב הסימולציה מאפשרת ייעול של העבודה, אין צורך בהתמסחות בין כלים שונים, שפות שונות ועוד. כמו כן, עבודה עם כל מידול שיודיע לבצע גם את השירותים עצם ייעיל ויחסון זמן יקר בשירות כפול של תרשימי המודלים השונים בפרויקט.

3. העבודה עם כל מידול אשר מקנה לנו יכולת הפקט דוחות ובנית ארכיטקטורה ברמה גבוהה עוזר מאוד ומיעיל מאוד את התהליך, נותן לנו לראות בבירור את כל הקישורים השונים, בכל היררכיות השונות, תוך שימוש בפילטרים לפי הצורך, דבר אשר מפשט מאוד את ההסתכבות על השירותים.

- ביום נושא הפקט דוחות מtopic כל המידול הופך יותר ויותר לסטנדרט בתעשייה וזאת מכיוון שהוא יוצר סטנדרט אחיד המשקף את כל מערכות הפרויקט וכן כמו כן, הוא מייצר דוח אשר ניתן לשולח ולשתף עם כל בעלי העניין בפרויקט.
- כמו כן, הוא הפקט דוחות אוטומטיים עתידיים להחליף את כתיבת המסמכים השונים העוסקים בדוחות אלה, דבר אשר מסרב ומעקב לעיתים את מהלך הפרויקט.

4. יש לבצע כמה שיותר סימולציות, כמה שיותר מקיפות, אם ישנה אפשרות לאוטומציה, רצוי מאוד להשתמש בה, בפרויקט שלנו ביצענו בדיקות ידניות ואוטומטיות בכל.

.5. לכלי System Composer, אשר כМОצג בפרקים הקודמים שימש לביצוע סימולציה, ישנים יכולות הקלטה לצורך חקירה לאחרור, עקב גודל הממחקר והזמן המוגבל תכונה זאת רק הדוגמה לנו ולא נבחנה על ידנו אך נראהית כתכונה חשובה ו שימושית.

## 5.2. לוחים עיקריים

בנוסף למסקנות שהוצעו בסעיף 5.1 לעיל המתייחסות לשאלות המחקר שהוגדרו בתחילת העבודה ונחקרו במהלך, ברכוננו להוסיף לוחים מהעבודה שבוצע במהלך זה, שנדראים לנו חשובים למהנדסי מערכות בכל יישום:

1. לא להסתמך על שרטוט בלבד, בהתחלה השקענו זמן רב ומאזים על מנת להגיע לשרטוטים יפים ועמוקים ככל האפשר, הדבר גזל זמן רב מזמן הפרויקט, שהמשך הסתר כזמן שלא נוצל ראוי. לאחר שהתחלנו להשתמש בכלים המידול הבנו שנראות הארכיטקטורה בתוך הכלים ברורה יותר שכן כל מהנדסי המערכת מכירים ועובדים עם הכלים ולכן עדיף השקיע את המאמץ מלבתיחה בבניית הארכיטקטורה בכלים המידול.
2. הפקת דוחות (בתחילת אמצע ובסוף הפרויקט) מתוך כלים המידול והסימולציה היא יכולה להיות מאד יעילה. יכולת זו מאפשרת שיתוף של הארכיטקטורה גם למי שאין את כלים המידול ומעלה הכל יכולת לבדוק בביטחון שונות ובעצם לחסוך הרבה זמן שמתבצע על תכנון כפול בכלים השertos השונים (כגון: Visio) אשר גם ככא נבנים מכלים המידול והסימולציה, שכן לאחר זהוי הבעיה בכלים המידול והסימולציה חלק נכבד מהארכיטקטורה נבנה מחדש.
3. ביצוע סימולציות רבות ככל האפשר לפני תחילת פיתוח הפרויקט אמיתי, ככל שהתקדמנו בסימולציות הפרויקט נוכחנו לדעת שישנים הרבה שינויים, תוספות ואפילו בניהיה חדש של הארכיטקטורה. הדבר היה מקנה עבורנו חיסכון עצום בעלות אם אכן היו צריכים לאחר תחילת שלב פיתוח אמיתי למוצר.

### 5.3. מגבלות המחבר

1. זמן – עקב הזמן המוגבל, כמות הכלים הנחקרים הייתה גם היא מוגבלת ולכן בחרנו להתייחס ל-2 כלים אשר בשימוש נרחב בתעשייה עם השיטות המתאימות כפי שתואר בפרק 1

2. קישור המודל אל כלל המערכת –

לתוכנה MATLAB ישנו הרבה כלים, אחד מהם הוא ה- SC (System Composer), כלי מידול וסימולציה, בו הتمקדמו בפרויקט הנ"ל. את ה- SC ניתן לקשר אל שאר הכלים של MATLAB, כלים כמו ה-state flow, בניית מודלים פיזיקליים ועוד. כל כלי זהה הוא "עולם ומלאו", ומכיון שהמחקר מתעסק בהנדסת מערכות, והכלי בו הتمקדמו הוא SC, לא קישרנו את המודל אל החלקים הפיזיים שבו (כגון: גוף הרכיב, גוף הקרון ועוד)

3. לימוד הכליל System Composer אשר נמצא בשלבי פיתוח ושינויים רציפים היו עבוריינו מכשול, הן בהיבט התיאורטי והן בהיבט המעשי,  
a. בהיבט התיאורטי –

- i. היה חומר רב באתר של MATLAB, אך לא תמיד כולם היה רלוונטי, בעיקר עקב שינויים בין גרסאות התוכנה השונות.
- ii. בעיות אשר צצו לא תמיד התגלו לפני כן ולמעשה הינו הריאשוניים (בישראל ואולי בעולם) לחווות אוטם.
- iii. דרישות מהכליל אשר נראה טריומאליות עבוריינו אך עדין לא פותחו (כגון הפקת דוחות).

b. בהיבט המעשי –

- i. השימוש בתוכנה היה עמוק ומורכב יותר מהצפוי.
- ii. למידת כלל האפשרויות הנרחבות אשר קיימות לך זמן רב יותר מהצפוי.

4. קורונה – בעיצומו של המחקר החלה מחלת הקורונה והסגר, עקב לכך, ההנחיה על הכל SC, כלי מורכב מאוד, ללא אינטראקטיבית אישית הקשתה את תהליך הלמידה.

5. בזכות יכולות הסימולציה של הכלי System Composer, אחד הכלים הנחקרים, הארכיטקטורה והתרחישים וואפן פעולה החישנים שונים מספר פעמים וזאת בהתאם לביצוע הסימולציות השונות (המשמעות התחלה מחדש, לא ברור האם זאת מוגבלת של המחקר אבל ברצוינו לציין זאת, כמתואר במסקנות זה יכול דווקא להיות יתרון של מידול וסימולציה בפרויקט אמיתי לפני בניית המרכיבים, אבל בפרויקט מחקר גורם לחזרות רבות על להגעה למצאים)

#### 4.5.2. פתרונות המשך מומלצים

1. המשך הפרויקט ו קישורו אל החלק הפיזי של המערכת
2. הוספת חישנים וחלקים נוספים
3. ניסיון ביצוע פעולות אוטומטית באמצעות Scripts הן לסימולציה מערכית והן להפקת דוחות
4. ביצוע סימולציות בכלים נוספים כגון Enterprise Architect והשוואה אל המחקר הנ"ל
5. השוואת כלי מידול/شرطוט נוספים מקוד פתוח (כגון: ArgoUML, BOUML) והשוואה אל כלי המידול/شرطוט במחקר זה. כמו כן, לדון בעניין בטחון המידע בשימוש בכלים מידול/شرطוט קוד פתוח.

## רשימת מקורות

1. ArgoUML - Open Source Unified Modeling Language (UML) Tool. Retrieved from <https://www.methodsandtools.com/tools/tools.php?argouml>
2. AUTOMATIC SPLIT AND COMBINE SYSTEM OF THE SHINKANSEN TRAIN. Retrieved from <https://www.sciencedirect.com/topics/engineering/speed-sensor>
3. Banas, T. (2018, April 29). How to Calculate Crash Forces. Retrieved from <https://sciencing.com/calculate-crash-forces-6038611.html>
4. Bonnici, D. (2020, May 25). 360-degree parking monitors explained. Retrieved from <https://www.whichcar.com.au/car-advice/360-degree-parking-monitors-explained>
5. David Harel. (2007). Statecharts in the Making: A Personal Account. *The Weizmann Institute of Science*.
6. Enterprise Architect 15.1 User Guide. Retrieved from [https://sparxsystems.com/enterprise\\_architect\\_user\\_guide/15.1/index/index.html](https://sparxsystems.com/enterprise_architect_user_guide/15.1/index/index.html)
7. Faida Mhenni, Jean-Yves Choley, Olivia Penas, Regis Plateaux, Moncef Hammadi (2014). A SysML-based methodology for mechatronic systems architectural design. *Advanced Engineering Informatics Vol.28(3)*, P218-231

8. Feldman, Y. (2014, March). Conference: the 10th International Modelica Conference. Simulating Rhapsody SysML Blocks in Hybrid Models with FMI. Retrieved from  
[https://www.researchgate.net/publication/269235773\\_Simulating\\_Rhapsody\\_SysML\\_Blocks\\_in\\_Hybrid\\_Models\\_with\\_FMI](https://www.researchgate.net/publication/269235773_Simulating_Rhapsody_SysML_Blocks_in_Hybrid_Models_with_FMI)
9. Friedenthal, S., Moore, A., Steiner, R. (2015) A Practical Guide to SysML: The Systems Modeling Language. Retrieved from
10. Global Positioning System. (2010). Retrieved from  
<https://www.sciencedirect.com/topics/engineering/global-positioning-system>
- . Hause, M & Stuart, A & Richards, D & Holt, J. (2010). Testing Safety Critical 11 Systems with SysML/UML. *2010 15th IEEE International Conference on Engineering of Complex Computer Systems*, P.325-330
12. Holt.J. , Perry,S. (2008) SysML for Systems Engineering. Retrieved from  
[https://books.google.co.il/books?hl=iw&lr=&id=OEKtufR7spYC&oi=fnd&pg=PA1&dq=sysml&ots=LpcCmwH6Bq&sig=oHuGZ8T95-GNMvoQ2MZ3FJgSH3s&redir\\_esc=y#v=onepage&q=sysml&f=false](https://books.google.co.il/books?hl=iw&lr=&id=OEKtufR7spYC&oi=fnd&pg=PA1&dq=sysml&ots=LpcCmwH6Bq&sig=oHuGZ8T95-GNMvoQ2MZ3FJgSH3s&redir_esc=y#v=onepage&q=sysml&f=false)
13. How to Measure Pressure with Pressure Sensors. (2019, Mar 14). Retrieved from  
<https://www.ni.com/en-il/innovations/white-papers/06/how-to-measure-pressure-with-pressure-sensors.html>
14. INCOSE. Retrieved from <https://www.incosse.org/>

15. Kai Yi, W. (2019, July 10). How does a 360-degree parking camera work? . Retrieved from <https://www.torque.com.sg/features/how-does-a-360-degree-parking-camera-work/>
  
16. Kyes, J. (2020, May 22). GPS, or the Global Positioning System, is a global navigation satellite system that provides location, velocity and time synchronization. Retrieved from <https://www.geotab.com/blog/what-is-gps/>
  
17. Lavi, Jonah Z. & Kudish, Joseph. (2005). Systems modeling & requirements specification using ECSAM: an analysis method for embedded & computer-based systems. *Innovations in Systems and Software Engineering, Vol.1(2)*, P100-115
  
18. Lavi, J.Z & Kudish, J. (2000). Systematic derivation of operational scenarios using ECSAM. *Proceedings Seventh IEEE International Conference and Workshop on the Engineering of Computer-Based Systems*, P.359-366
  
19. Ley, M. (2019, October). Our Journey Towards Model-Based Product Lines. Retrieved from <https://uk.mathworks.com/videos/our-journey-towards-model-based-product-lines-1573233985120.html>
  
20. List of Unified Modeling Language tools. Retrieved from [https://en.wikipedia.org/wiki/List\\_of\\_Unified\\_Modeling\\_Language\\_tools](https://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools)
  
21. Mangeruca, L. (2012, May 30). Simulation framework specification D\_7.1.1. Retrieved from <http://danse-ip.eu/home/>

22. Mert, V. (2019, November). Architectural Composition of Models by Simulink® System Composer. Retrieved from <https://vtol.org/store/product/architectural-composition-of-models-by-simulink-system-composer-15308.cfm>
23. Moving Visio Diagrams into Enterprise Architect. Retrieved from <https://sparxsystems.com/products/visio-importer/moving-from-visio-to-enterprise-architect.html?ref=eaRes>
24. Ober, Ileana & Ober, Iulian & Dragomir, Iulia & Aboussoror, El. (2011). UML/SysML semantic tunings. *Innovations in Systems and Software Engineering*, Vol.7(4), P.257-264
25. Olivier Casse. (2017). SysML: Object Management Group (OMG) Systems Modeling Language. *SysML in Action with Cameo Systems Modeler*.
26. OMG Systems Modeling Language (OMG SysML™), Version 1.3. (2012, June). Retrieved from <https://www.omg.org/spec/SysML/1.3/PDF>
27. Pearce, P. & Friedenthal, S. (2013). A Practical Approach For Modelling Submarine Subsystem Architecture In SysML. *Submarine Institute of Australia Science, Technology & Engineering Conference*, P 347
28. Professor Lehman, R. The Mechanical Properties of Glass. Retrieved from <http://glassproperties.com/references/MechPropHandouts.pdf>
29. Reader on GPS positioning. Retrieved from <https://www.tudelft.nl/en/ceg/about-faculty/departments/geoscience-remote-sensing/education/bsc-education/reader-on-gps-positioning/>

30. Samuel, J. (2019, September 22). Documentation-driven Development. Retrieved from <https://medium.com/swlh/documentation-driven-development-f9a6d3258e5>
31. SCADA. Retrieved from <https://en.wikipedia.org/wiki/SCADA>
32. Shipton, L. (2017). Connecting System Architecture to Model-Based Design. Retrieved from [https://www.incose.org/docs/default-source/default-document-library/mathworks\\_modelbasedsystemsengineering.pdf?sfvrsn=6d549fc6\\_0](https://www.incose.org/docs/default-source/default-document-library/mathworks_modelbasedsystemsengineering.pdf?sfvrsn=6d549fc6_0)
33. System Composer. Retrieved from <https://www.mathworks.com/products/system-composer.html>
34. SysML Modelling: Block Definition Diagram (bdd). Retrieved from <https://www.threessl.com/pages/reference/diagrams/sysml-block-definition-diagram.php>
35. SysML Modelling: Internal Block Diagram (ibd). Retrieved from <https://www.threessl.com/pages/reference/diagrams/sysml-internal-block-diagram.php>
36. Transentis Consulting. Model-Driven Documentation. Retrieved from <https://www.transentis.com/current-topics/model-driven-documentation/>
37. Visio. Retrieved from <https://www.microsoft.com/en-us/microsoft-365/visio/flowchart-software>

## נספחים

פרק זה יוצגו הנספחים – הטעיפים ליד הכותרות מייצגים את הטעיף אליו שייך הנספה

### נספח א: סקירת החישנים (מתיחס לטעיף 2.5)

#### א-1 – חישון לחץ (מתיחס לטעיף 2.5.1)

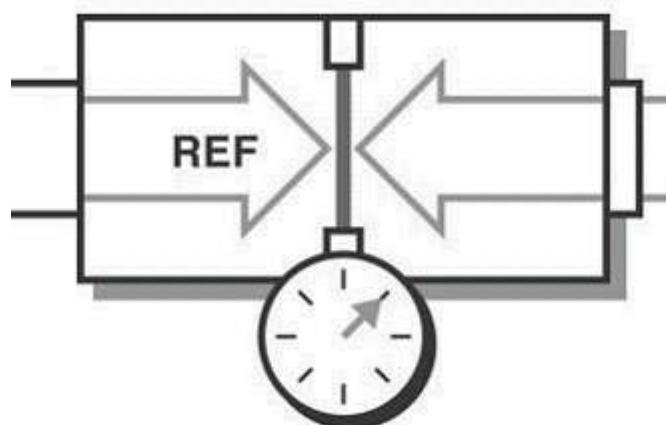
ע"פ המאמר (How To Measure Pressure with Pressure Sensors (2015

לחץ מוגדר ככוח לכל שטח יחידה שמופעל על סבבתו.

לדוגמה, לחץ  $P$  הוא פונקציה של כוח ( $F$ ) ושטח ( $A$ ):

$$P = F/A$$

החישון ימדד לחץ דיפרנציאלי, כאשר נבדוק מדידות דיפרנציאליות היחס לחץ ייחוס ספציפי (זמן נעילת הרכיב).



חישון לחץ דיפרנציאלי

#### מדידת לחץ:

התפוקה הטבעית של מתמר לחץ היא מתח (0-5V או זרם A (4-20mA).

שתי התפוקות הללו הינן ליניאריות לאורך טווח העבודה של המתמר.

למשל 0V או 4mA כאשר מדידת לחץ היא 0, ו-5V או 20mA כאשר היחס הנמדד הוא לחץ המרבי.

לאחר קבלת מתח מדיד, הוא מומר ליחידות לחץ (PSI - Pounds per Square Inch) (ההמרה מתבצעת בחישון והפלט יהיה ביחידות לחץ).

לטובות כך משתמשים בנוסחת ההמרה הבאה:

$$\text{Pressure} = \left( \frac{C_{fs}}{V_{ex}} \right) \left( \frac{V_{meas}}{CF} \right), \text{ where}$$

$C_{fs}$  = Full Scale Capacity - the maximum pressure which the transducer should receive

$V_{ex}$  = Excitation Voltage - the recommended input voltage

$V_{meas}$  = Measured Voltage - the raw voltage returned by the sensor

$CF$  = Calibration Factor - the output of the transducer, usually expressed in mV per input V

## א-2 – חישון מהירות (מתיחס לסעיף 2.5.2)

על המאמר AUTOMATIC SPLIT AND COMBINE SYSTEM OF THE SHINKANSEN TRAIN

מהירות היא מידת תיאור קצר תנועתו של גוף במרחב - המרחק שהוא עבר ביחידות זמן.

$$V = \frac{S}{t} \quad \text{מהירות} = \frac{\text{מרחק}}{\text{זמן}}$$

חישון המהירות משתמש בעקרון החישה המגנטית בעלת הרתיעה המשתנה, לפיה גרעין מגנטי קבוע

עם חוט סליל מפותל סביבו, מותקן על מנשא הרכזת הנិיח מייצר שדה מגנטי החופף לטבעת מסתובבת.

הטבעת המסתובבת יכולה להיות מסווג בטבעת שניינים או בטבעת עם חריצים ומחוברת לרכזת הגלגל או לפיר ההינע.

בעזרת מהירות הסיבוב של הגלגל קבועים את תדריות האות המועבר ליחידת הבקרה האלקטרונית.

בזמן שהגלגל והמצלב מסתובבים, השינויים והפערים או הצלעות והחריצים של הטבעת עוברים דרך השדה המגנטי של החישון.

הסיליה העוטף את החרוט המגנטי מרגיש את העוצמה המשתנה של השדה המגנטי כאשר השינויים או הצלעות עוברות בכווי

השיטף וכן נוצר מתחת לסירוגין בסליל שתדרו פרופורציונאלי ל מהירות הגלגל המסתובב.

המתח מועבר ליחידת הבקרה בכל פעם שגלגל הקרון מסתובב, בלי קשר אם הבלתי מופעלים.

מיוזג ועיבוד אוטומטי חישון המהירות ע"י יחידת הבקרה מספקים מהירות ייחוס שהיא מהירות הקרון.

**א-3 – מצלמות 360 (מתיחס לסעיף 2.5.3)**

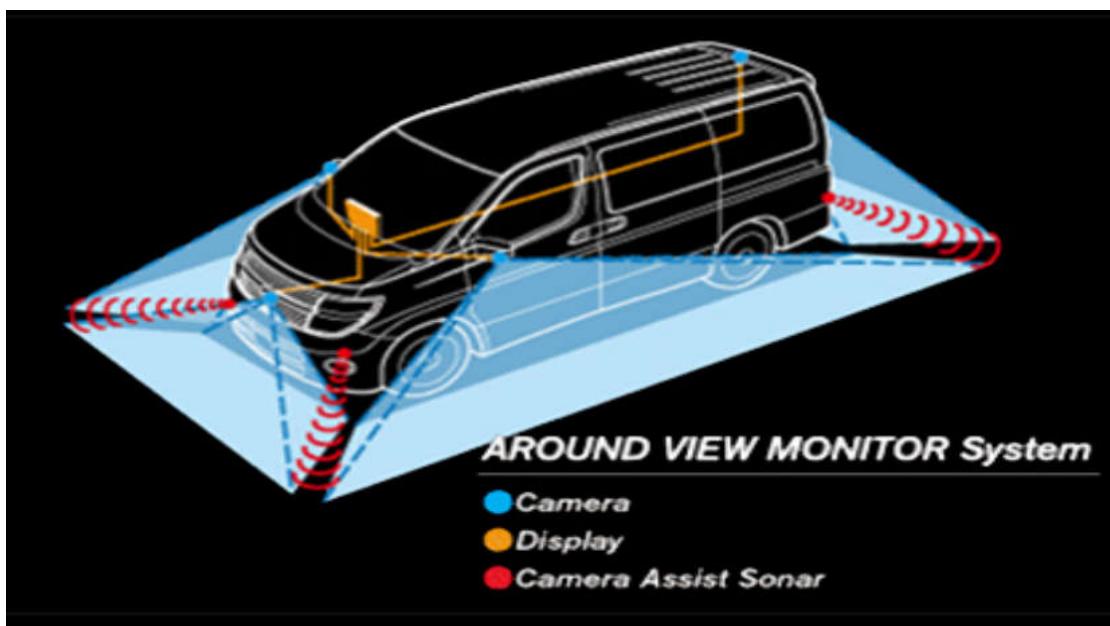
ע"פ המאמר 360 מעלות של מעלה הרכב ומה שמסביב (מכונה גם עין ציפור).

קליטה של 360 מעלות של מעלה הרכב ומה שמסביב (מכונה גם עין ציפור).  
מערכת המצלמות פועל ע"י תפירה ייחד של הזרות מהמצלמות מרחבי המכונית, ובכך מסוגל  
לספק

תצוגה הוליסטית של כל סביבת הרכב (360 מעלות).

במידת הצורך ניתן לקבל נתונים מצלמה אחת בלבד.

لمצלמות חדשות רוחבות במיוחד במילוי על מנת לתפוס את מרבית המידע הסובב את הרכב.



## א-4 – מקלט GPS (מתיחס לסעיף 2.5.4)

### כיצד ה-GPS עובד ?

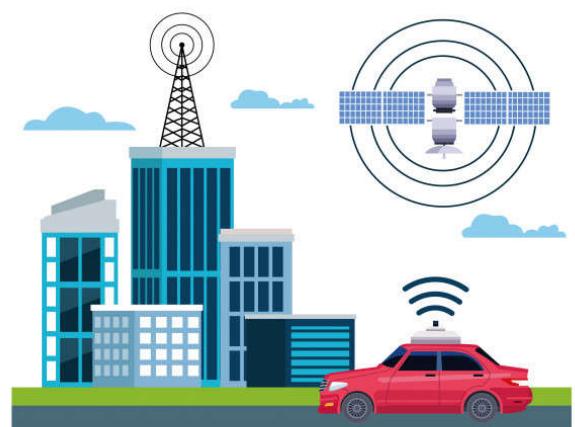
ע"פ המאמרים (Reader on GPS positioning, Global Positioning System (2010) **ו-** the Global Positioning System, is a global navigation satellite system that provides location, velocity and time synchronization

כשני תריסר לווייני GPS מקיפים את כדור הארץ בגובה 13000 מיילים בשיטה קבועות. הלוויינים אינם נייחים (ביחס לכדור הארץ), אלא נעים ממערב למזרח במהירות של כ-8000 קמ"ש ומקיפים את כדור הארץ פעמיים ביום.

**כל לוין מדוח ברכזיות:**

- Pseudorandom code – ID
- Ephemeris data - תאריך והשעה הנוכחיים והאם הלויין "בריא" או "לא בריא" ("לא בריא" לא בהכרח לוויין תקין, לפעמים מדובר על כיוול/הצבה מחדש).
- Almanac data - היקן הלויין נמצא.

המקלט ב- GPS לודד את זמן ההגעה (TOA) ואת זמן הטיסה (TOF) מהלוויין למקלט. בהתחשב בmphירות האור והיקן ומהירותו של הלוויין, המקלט GPS מחשב איפה נמצא המכוון. דרישים שלושה אותן GPS בצד לקבוע את מיקום המקלט ורביעי בצד לחשב גם את הגובה. ככל שהמקלט קולט יותר לוויינים כך איקונות האיכון משתפרת.

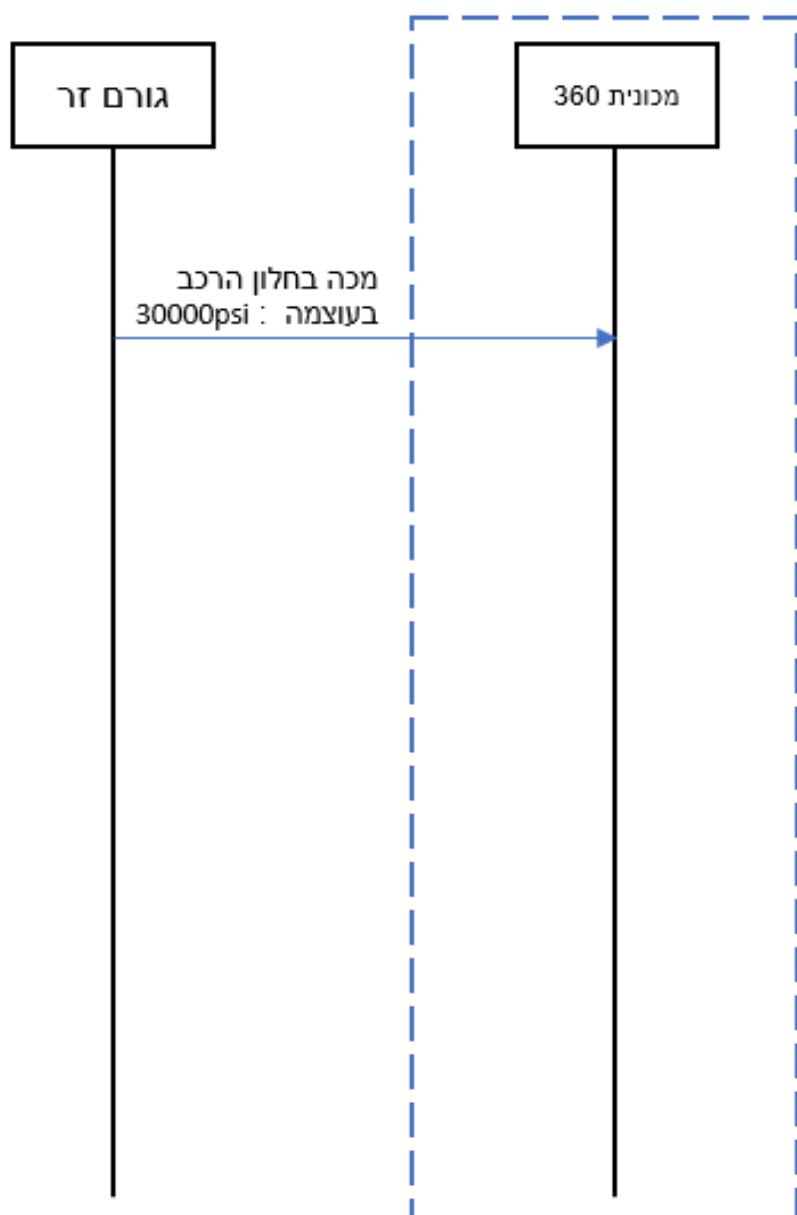


**נספח ב: תיאור התרחישים והמופעים (מתיחס לסעיף 3.2)**

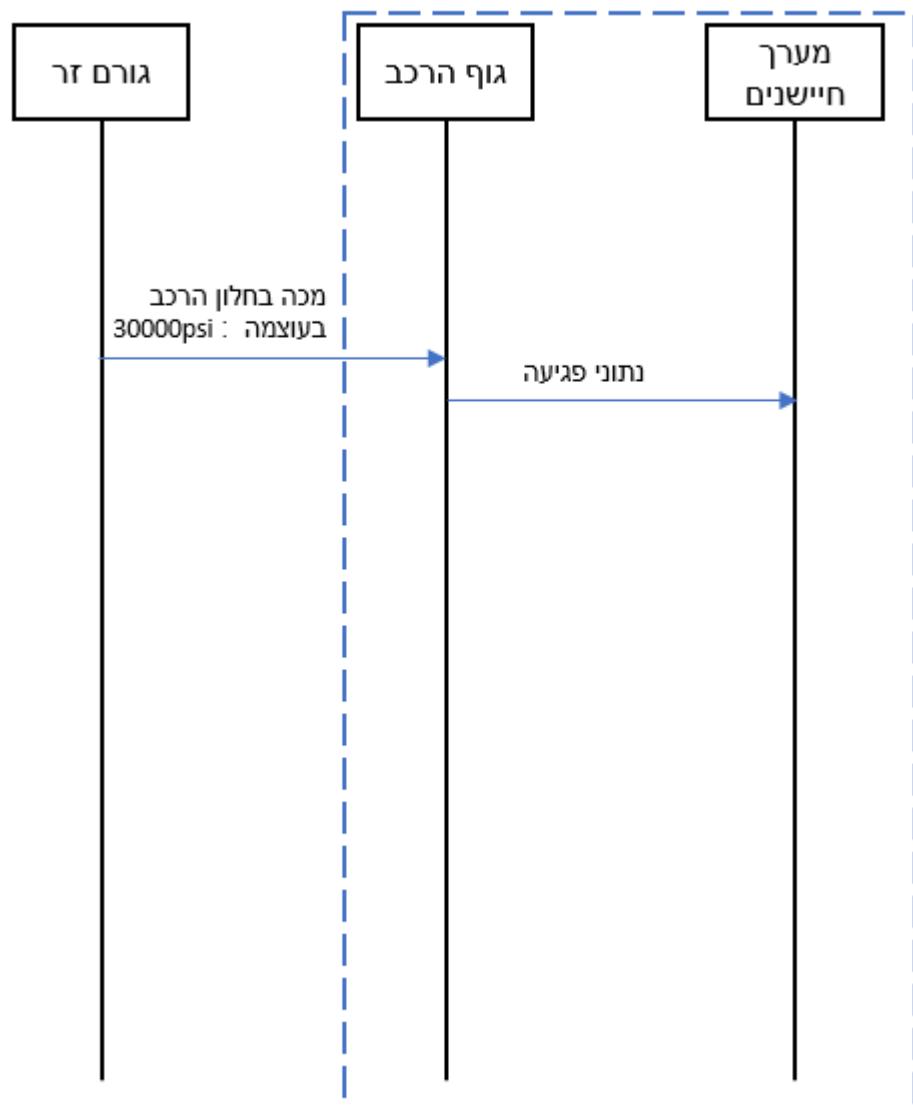
**ב-1 - ניטור פגיעה ברכיב 360: (מתיחס לסעיף 3.2.1)**

**ב-1.1 - תרחיש ניסיון פריצה על ידי פגיעה בחולון הרכיב:**

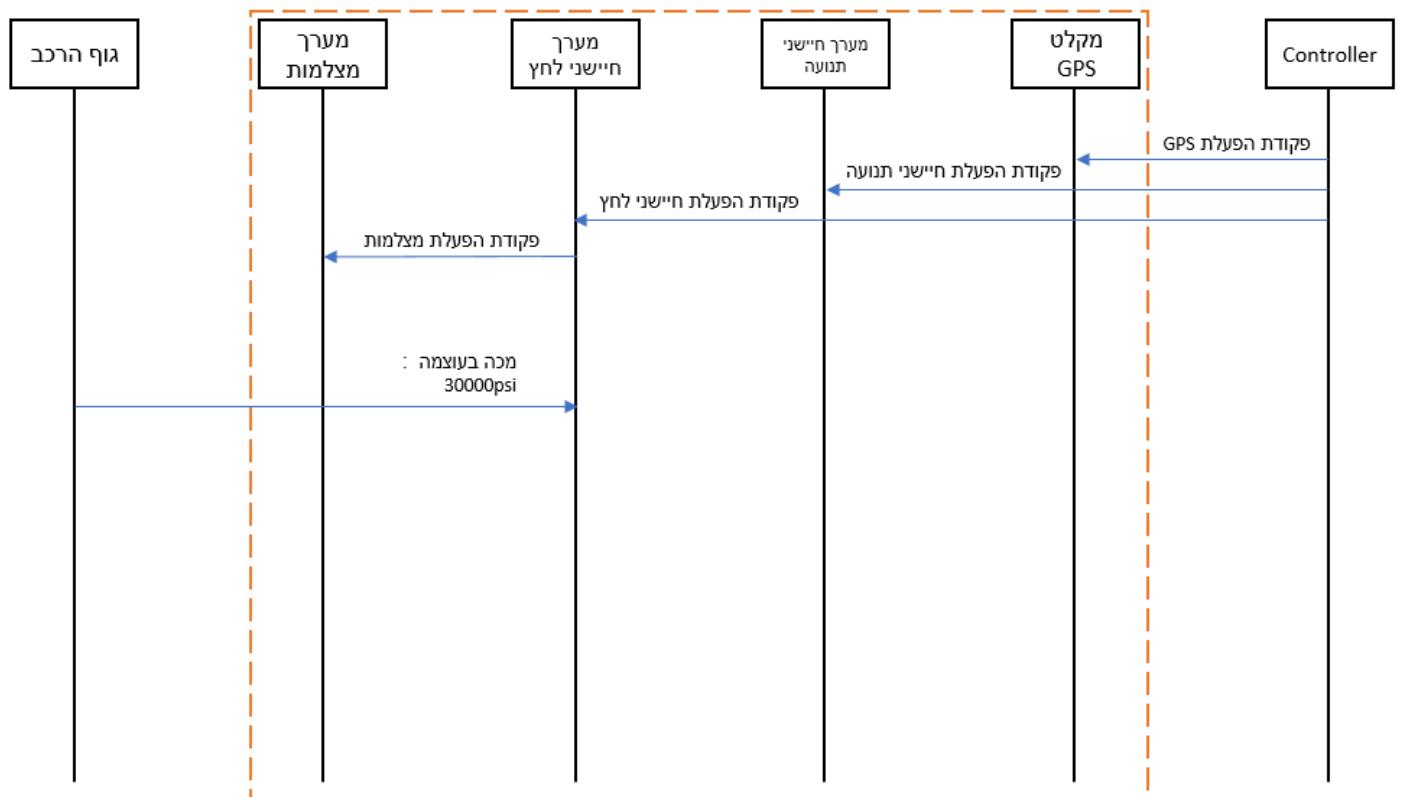
**ניסיון פריצה על ידי פגעה בחולון הרכיב – מופע 1 – קופסה שחורה:**



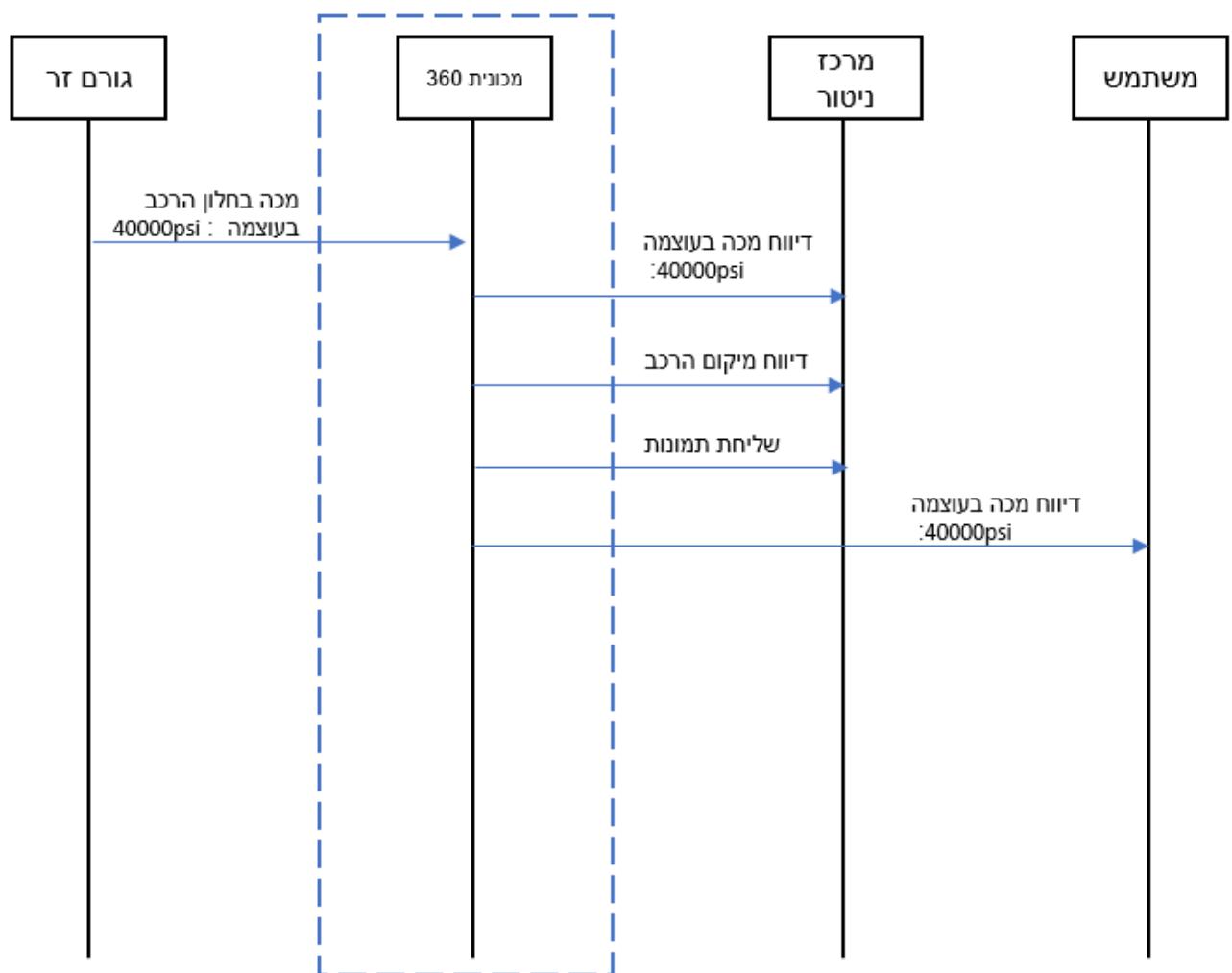
**ניסיון פריצה על ידי פגיעה בחילון הרכב – מופע 1 – קופסה לבנה (שכבה 1):**



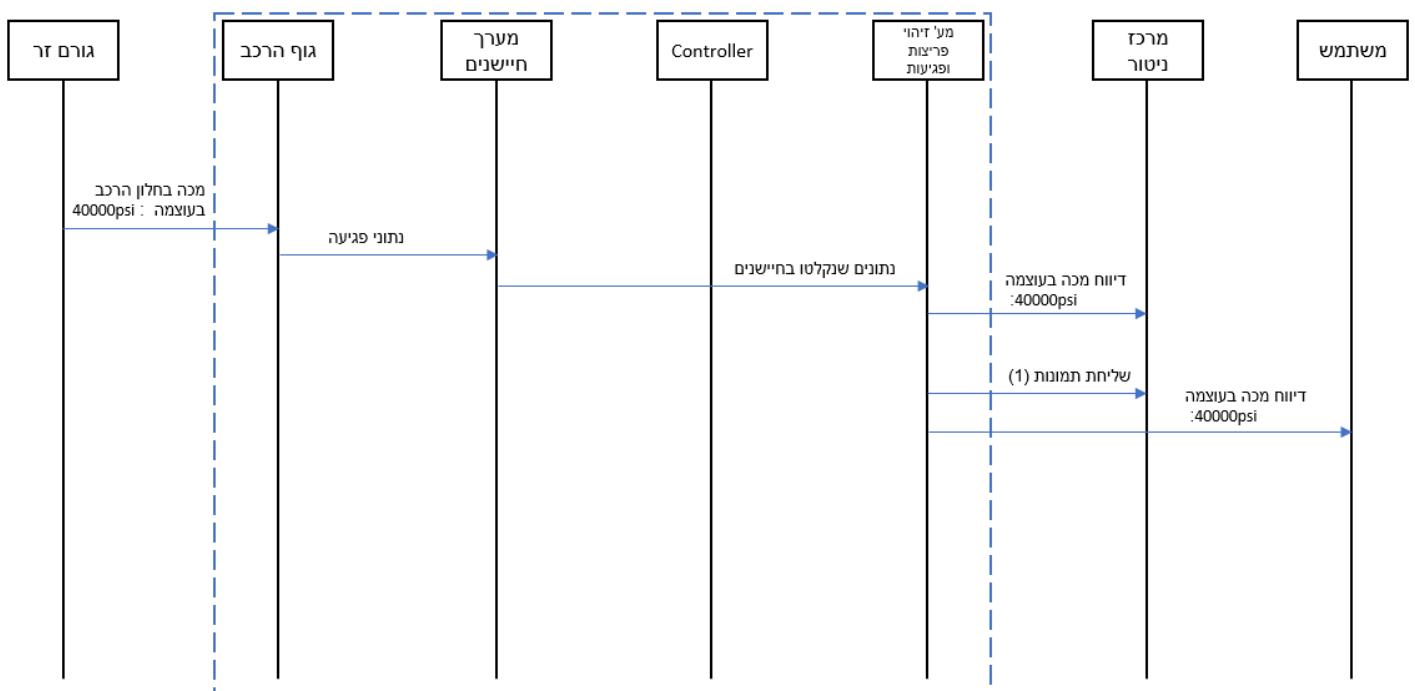
**ניסיון פריצה על ידי פגיעה בחלון הרכב – מופע 1 – קופסה לבנה (שכבה 2):**



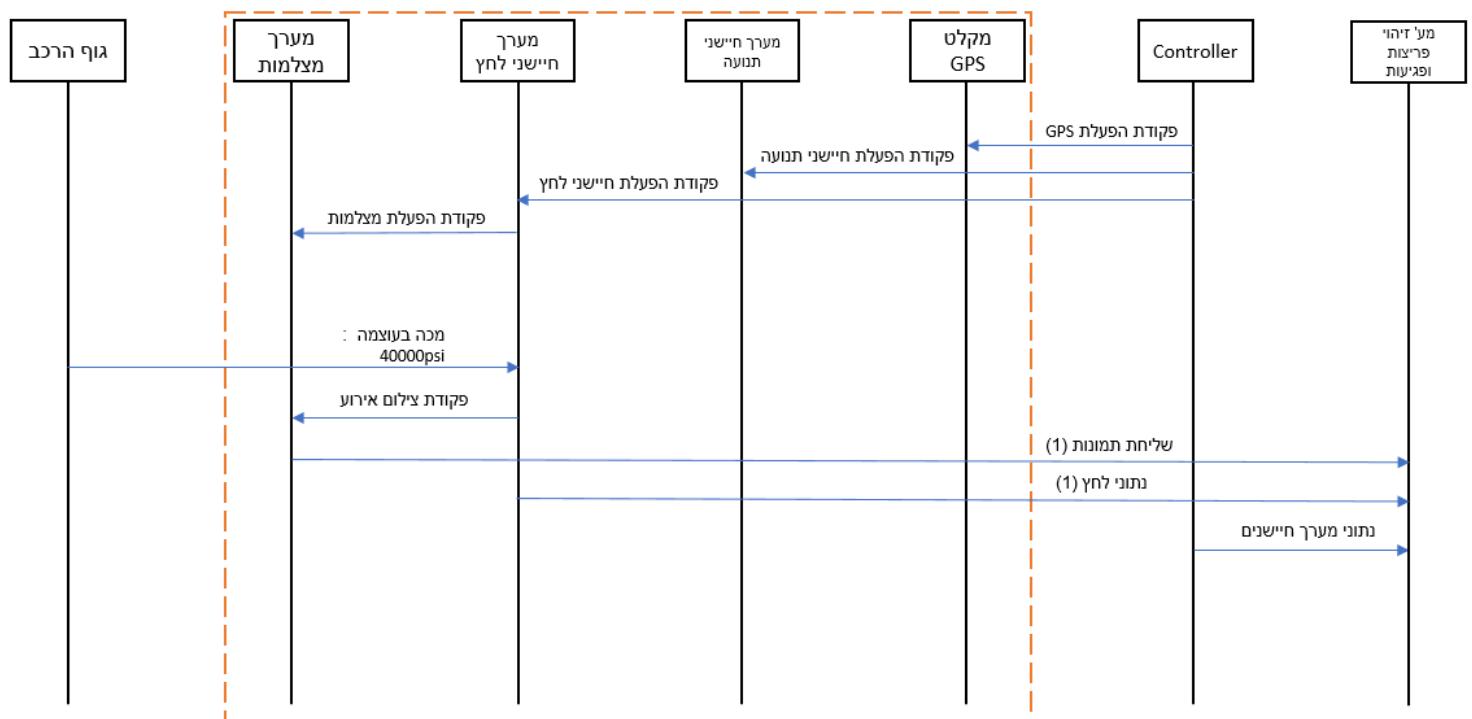
**ניסיון פריצה על ידי פגיעה בחלון הרכב – מופע 2 – קופסה שחורה:**



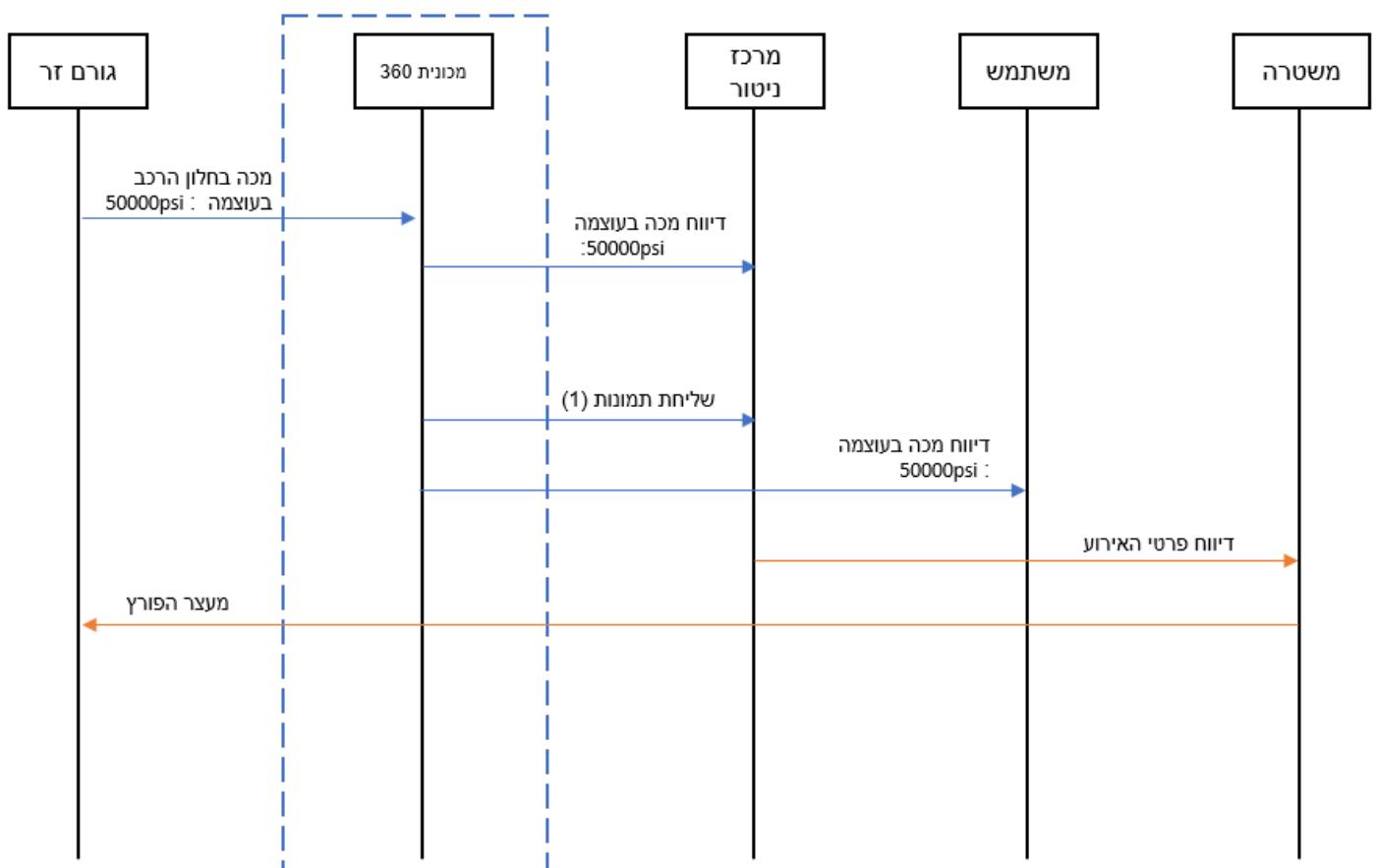
**ניסוי פגיעה על ידי פגיעה בחלון הרכב – מופע 2 – קופסה לבנה (שכבה 1):**



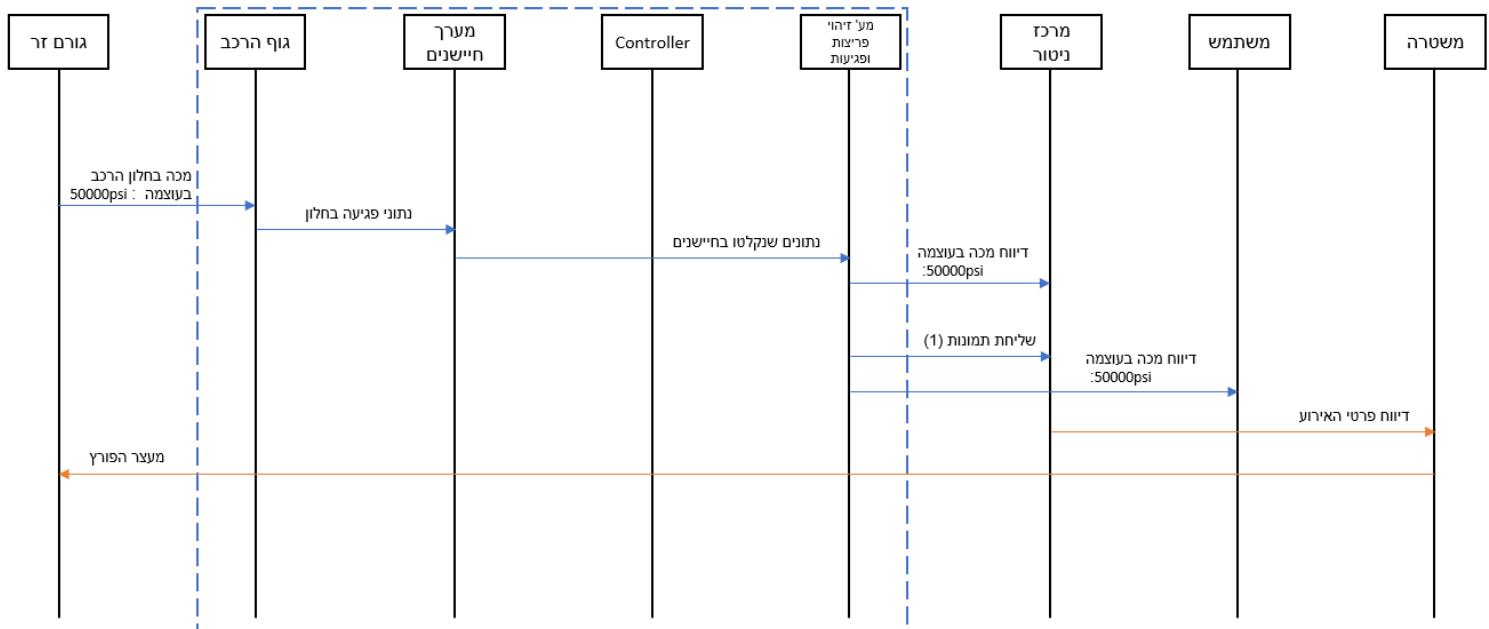
**ניסוי פריצה על ידי פגיעה בחלון הרכב – מופע 2 – קופסה לבנה (שכבה 2):**



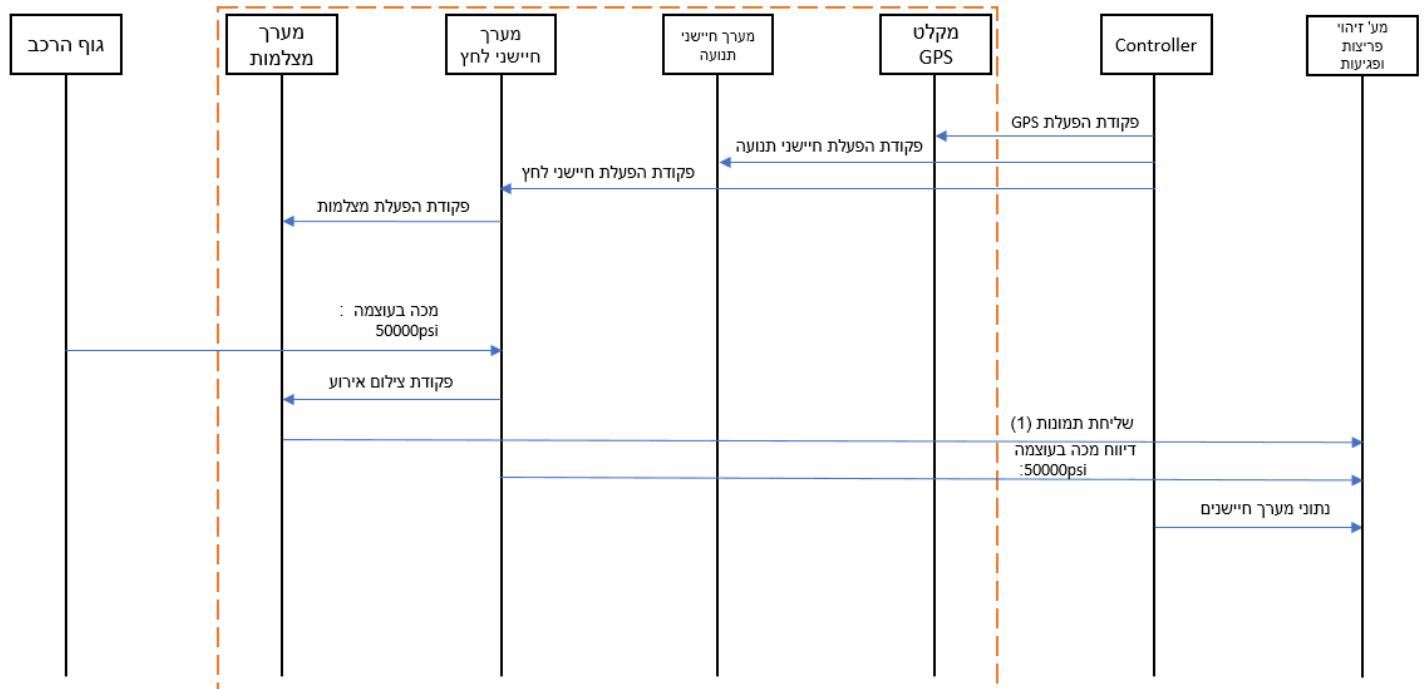
**ניסיון פריצה על ידי פגיעה בחלון הרכב – מופע 3 – קופסה שחורה:**



**ניסוי פגיעה על ידי פגיעה בחלון הרכב – מופע 3 – קופסה לבנה (שכבה 1):**

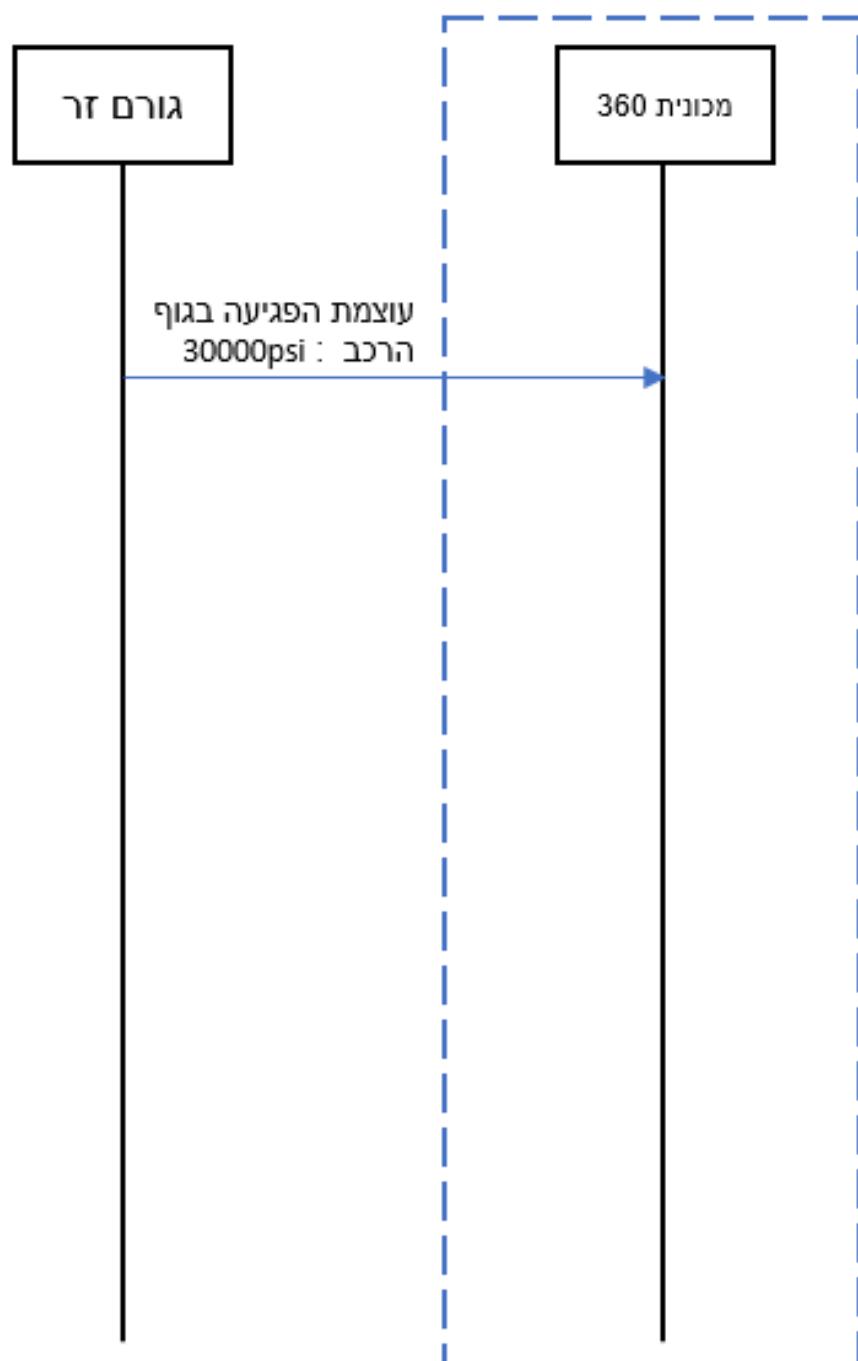


**ניסוי פגיעה על ידי פגעה בחלון הרכב – מופע 3 – קופסה לבנה (שכבה 2):**

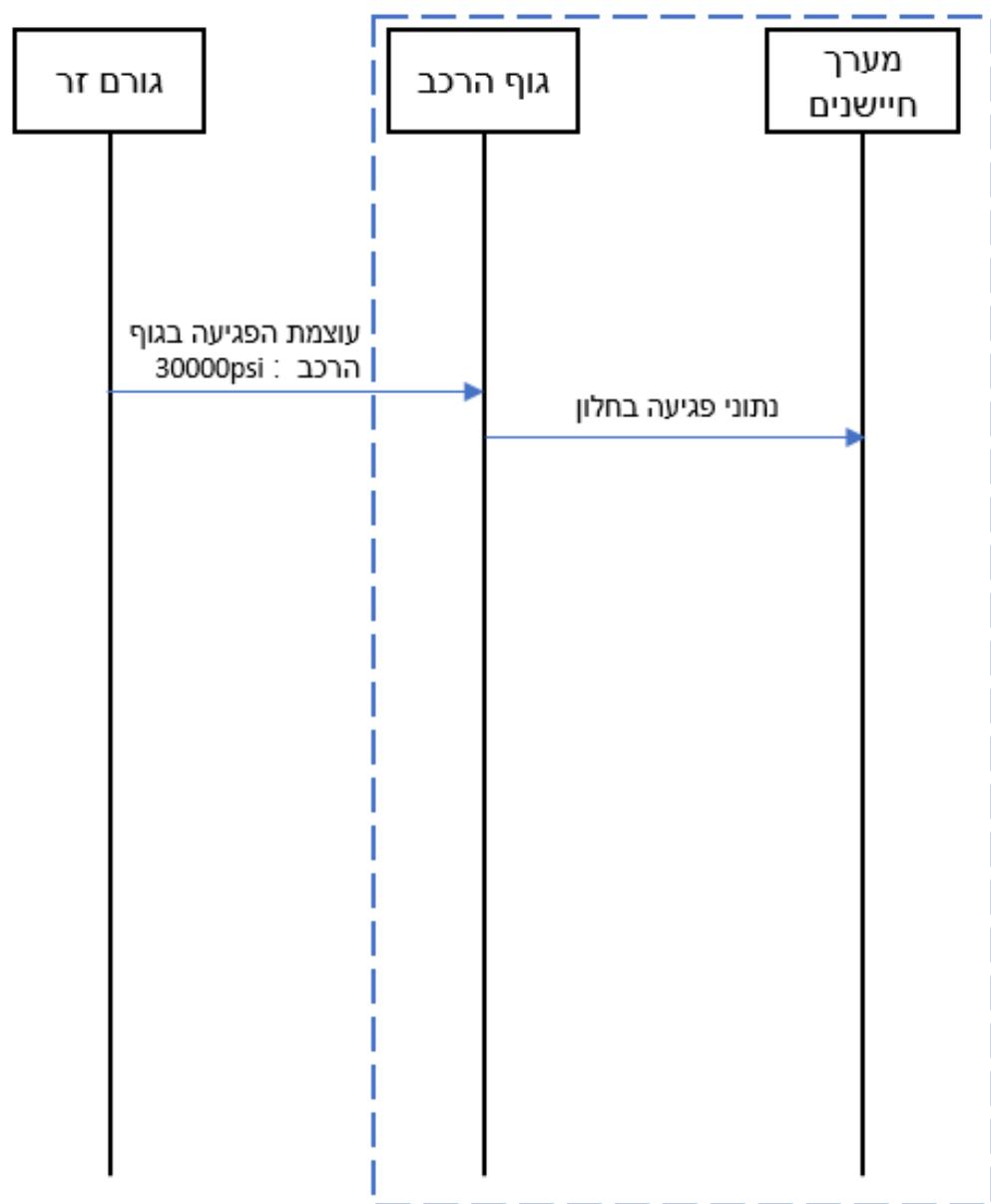


ב-2. – תרחיש תאונה - פגיעה בגוף הרכיב:

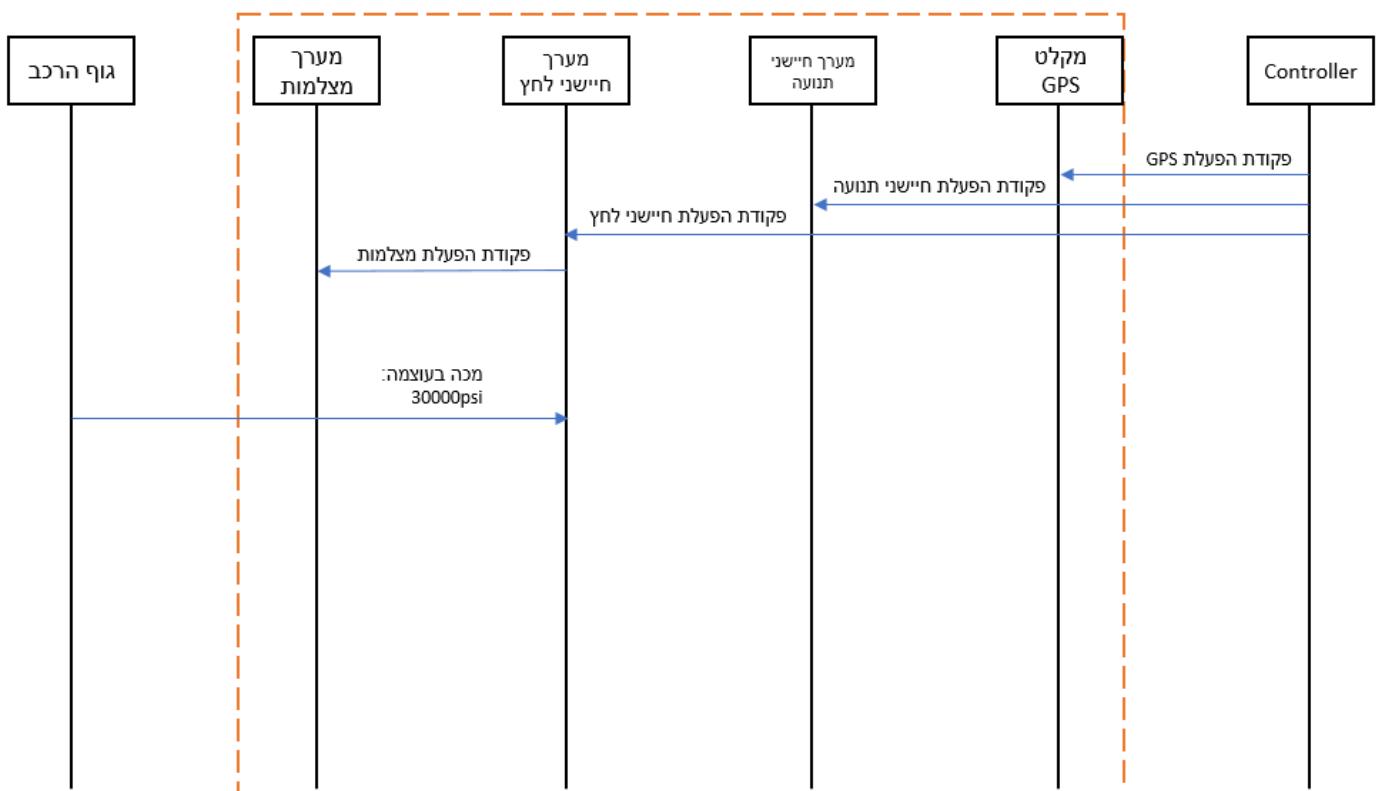
תאונה - פגיעה בגוף הרכיב – מופע 1 – קופסה שחורה:



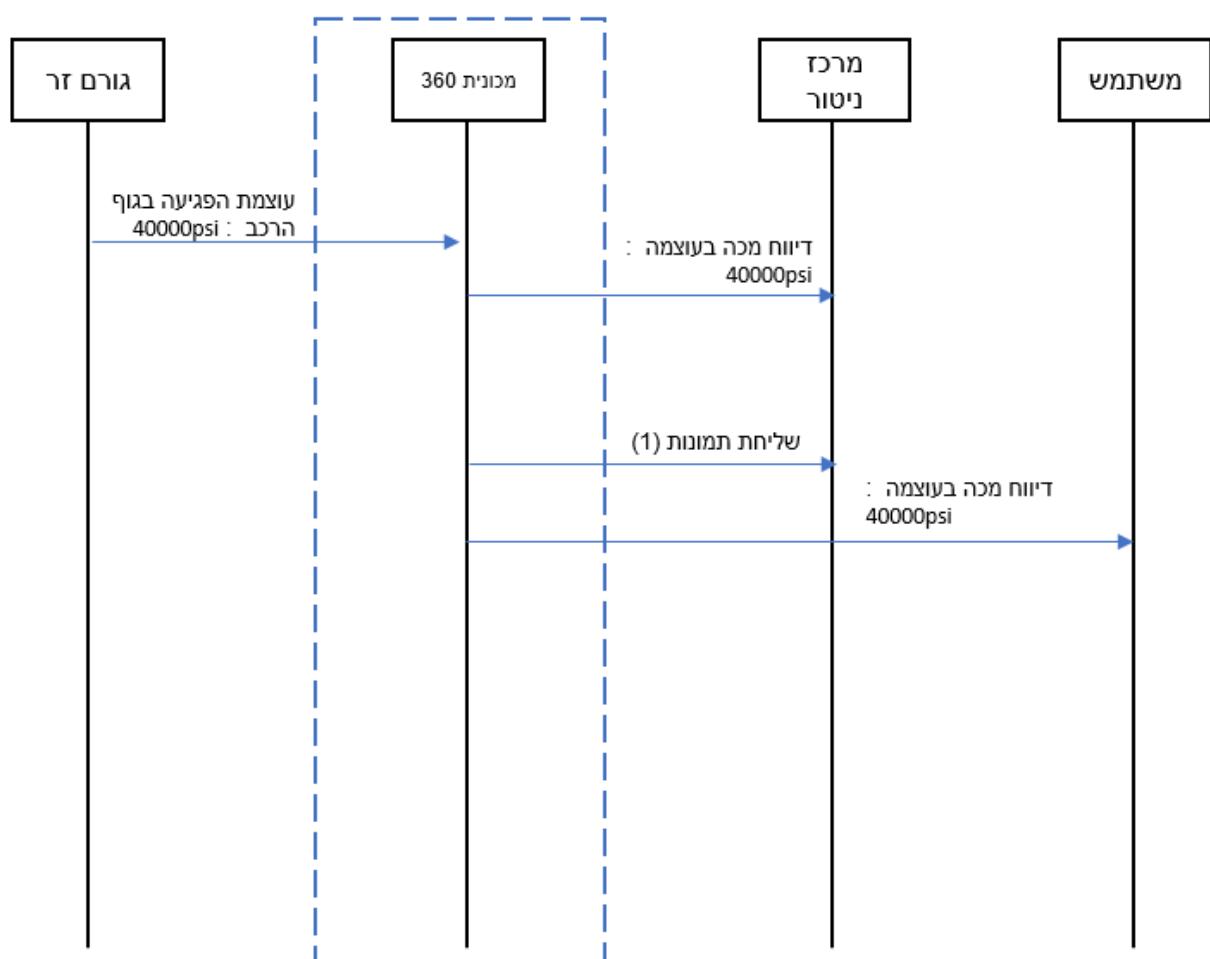
תאונה - פגיעה בגוף הרכב – מופע 1 – קופסה לבנה (שכבה 1):



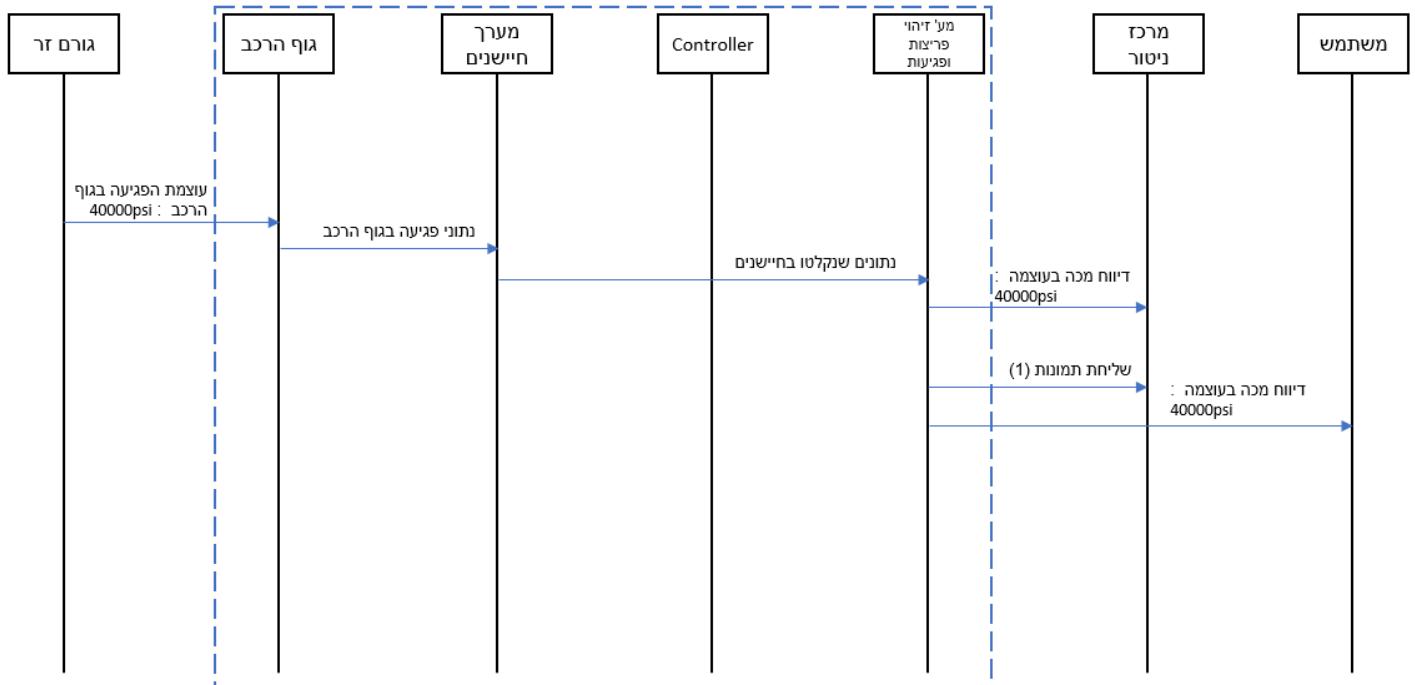
**תאונה - פגיעה בגוף הרכיב – מופע 1 – קופסה לבנה (שכבה 2):**



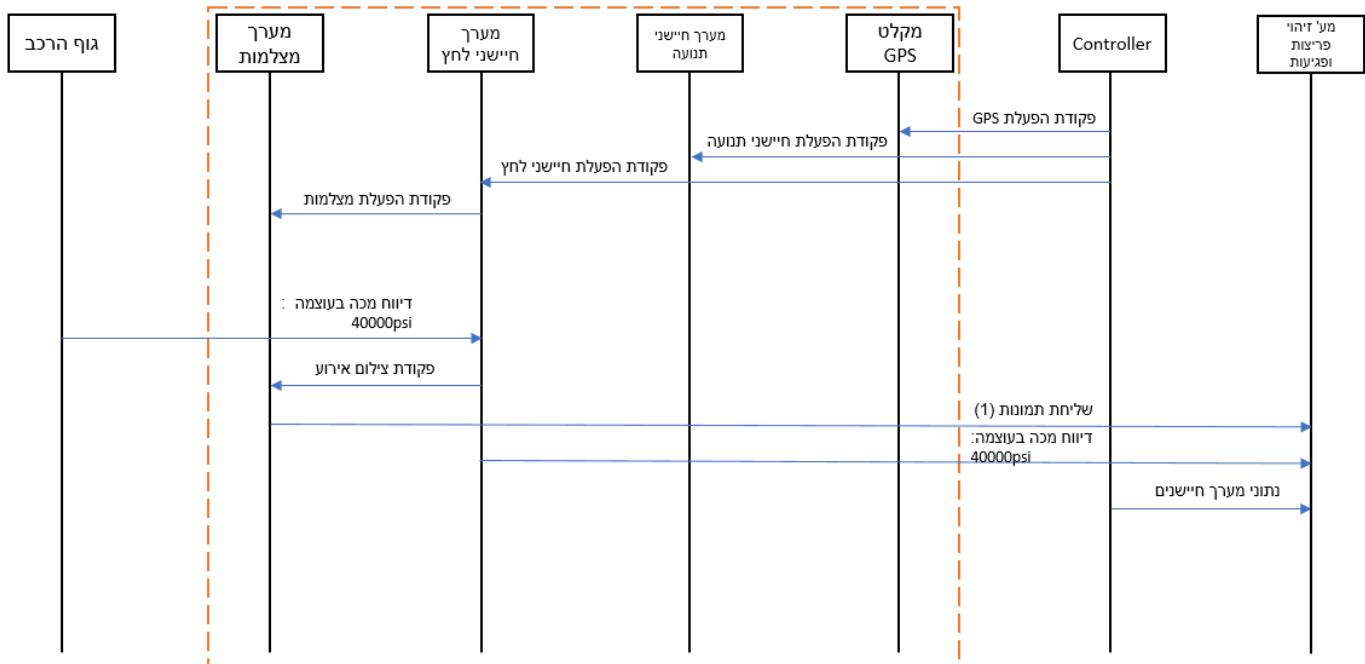
**תאונה - פגיעה בגוף הרכיב – מופע 2 – קופסה שחורה:**



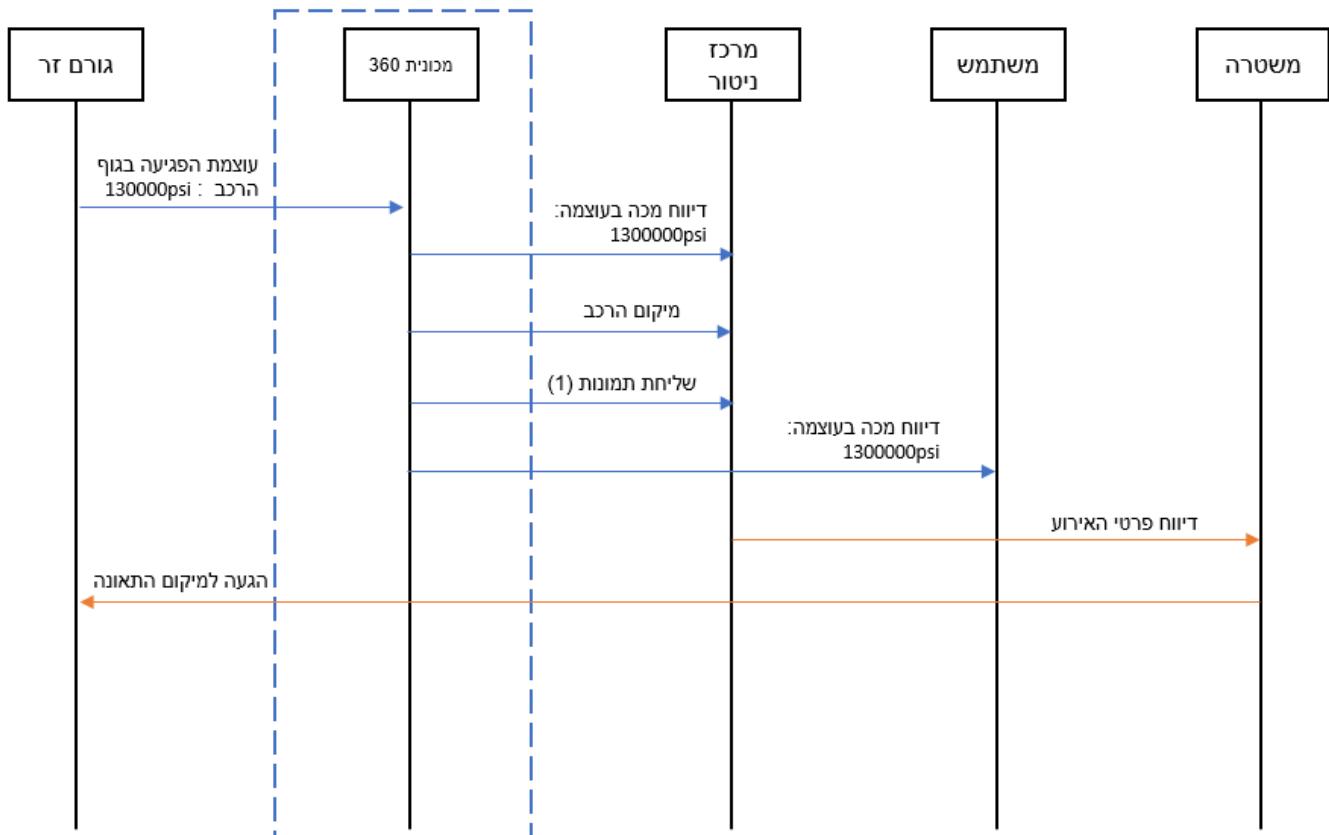
**תאונה - פגיעה בגוף הרכיב – מופע 2 – קופסה לבנה (שכבה 1):**



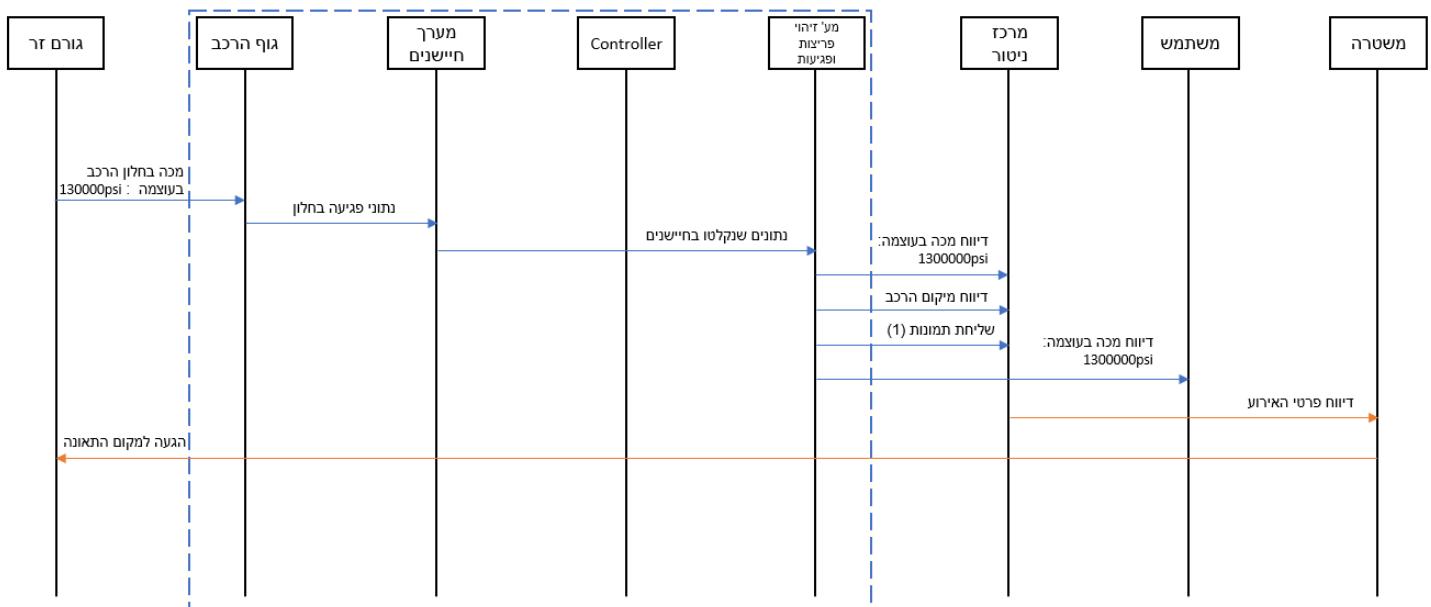
**תאונה - פגיעה בגוף הרכיב – מופע 2 – קופסה לבנה (שכבה 2):**



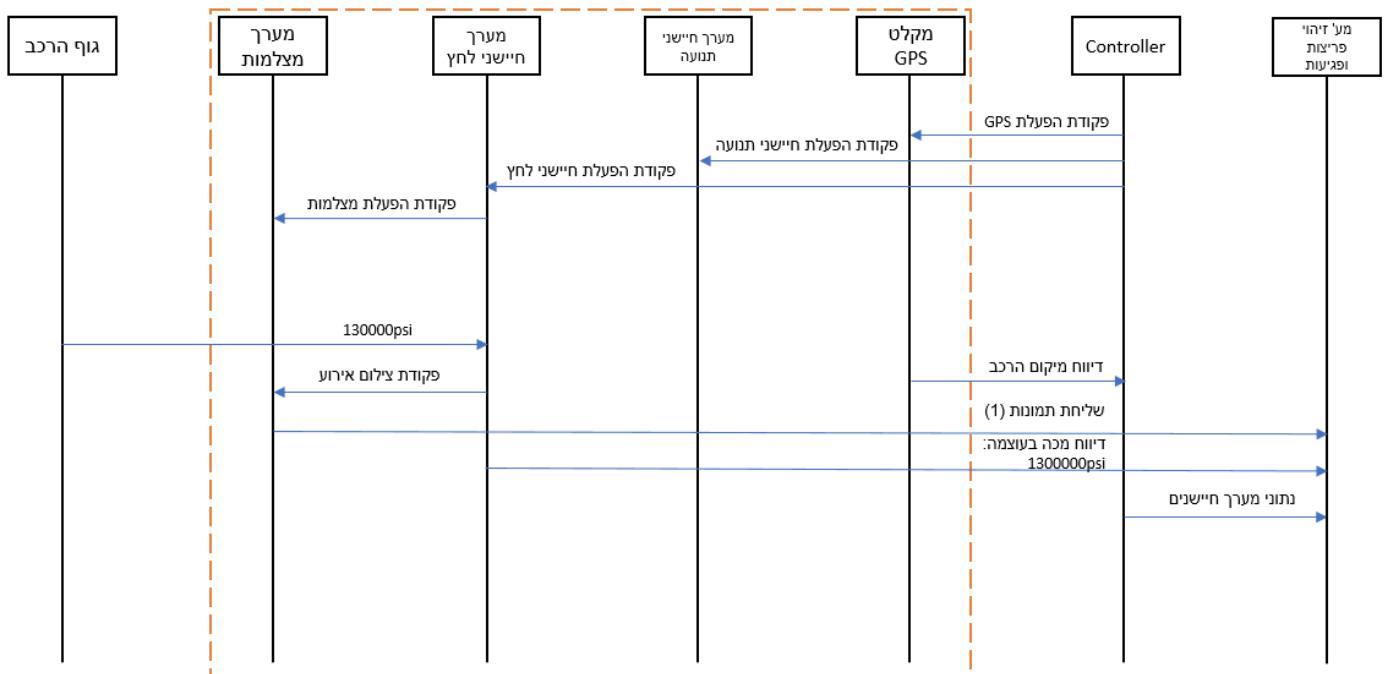
**תאונה - פגיעה בגוף הרכיב – מופע 3 – קופסה שחורה:**



**תאונה - פגיעה בגוף הרכיב – מופע 3 – קופסה לבנה (שכבה 1):**

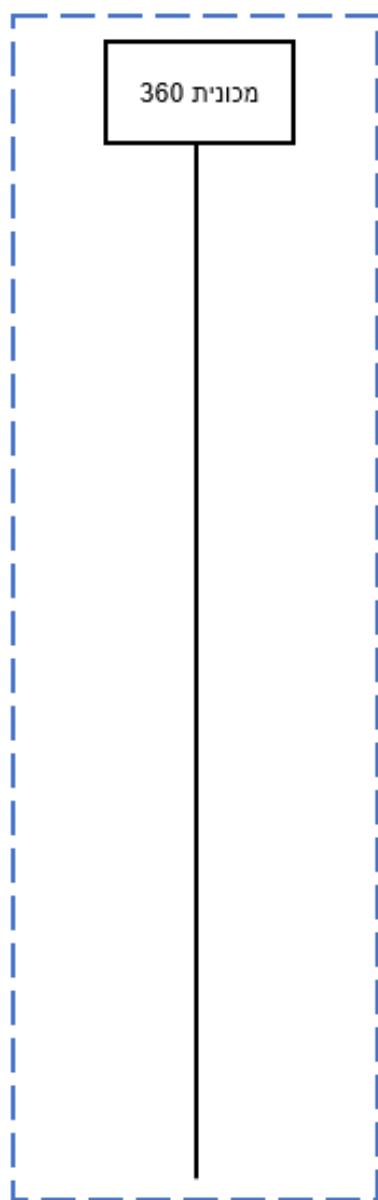


**תאונה - פגיעה בגוף הרכיב – מופע 3 – קופסה לבנה (שכבה 2):**

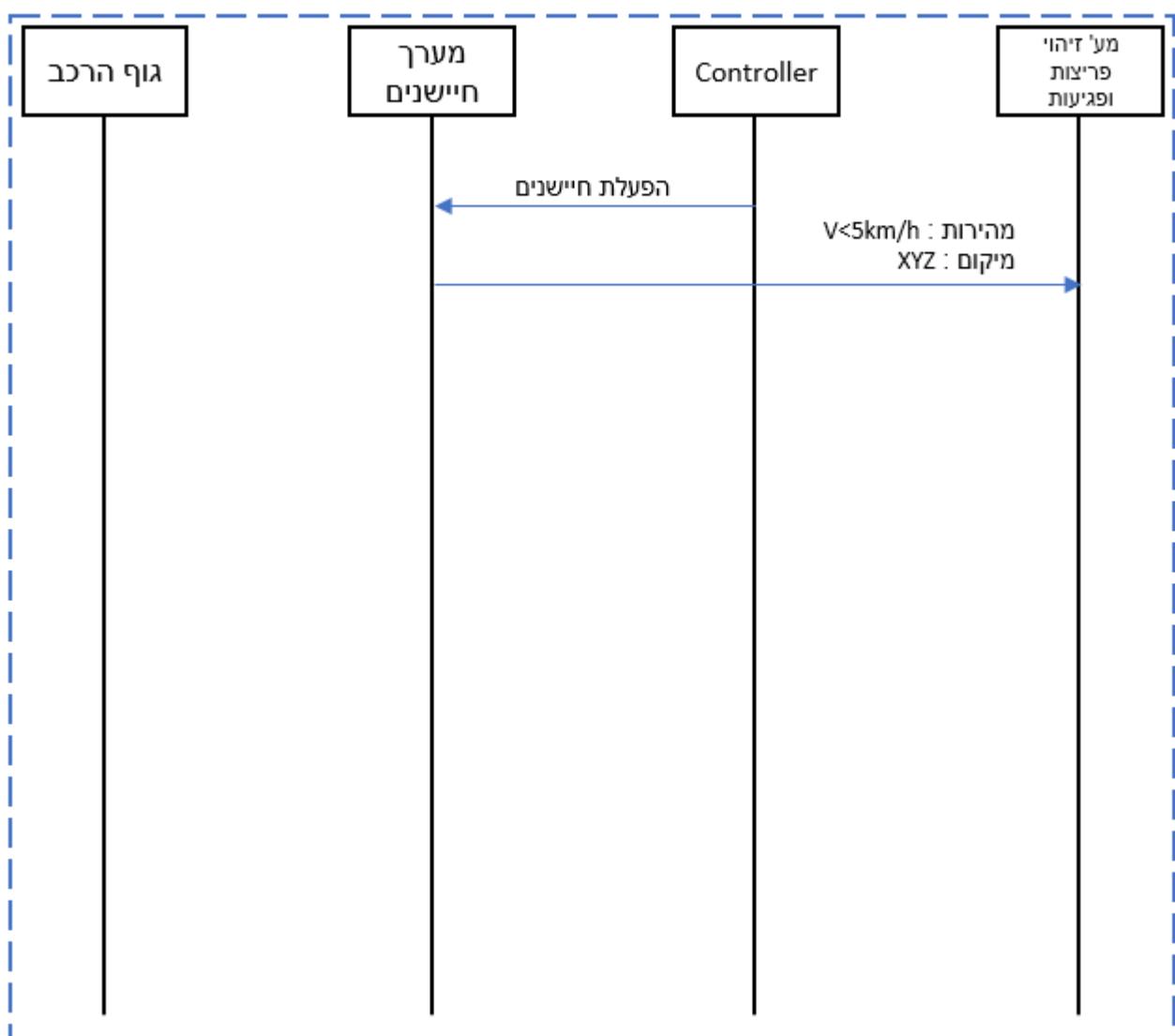


ב-1.3 – תרשים התדרדרות של הרכיב מהנניה בירידה (מצב N):

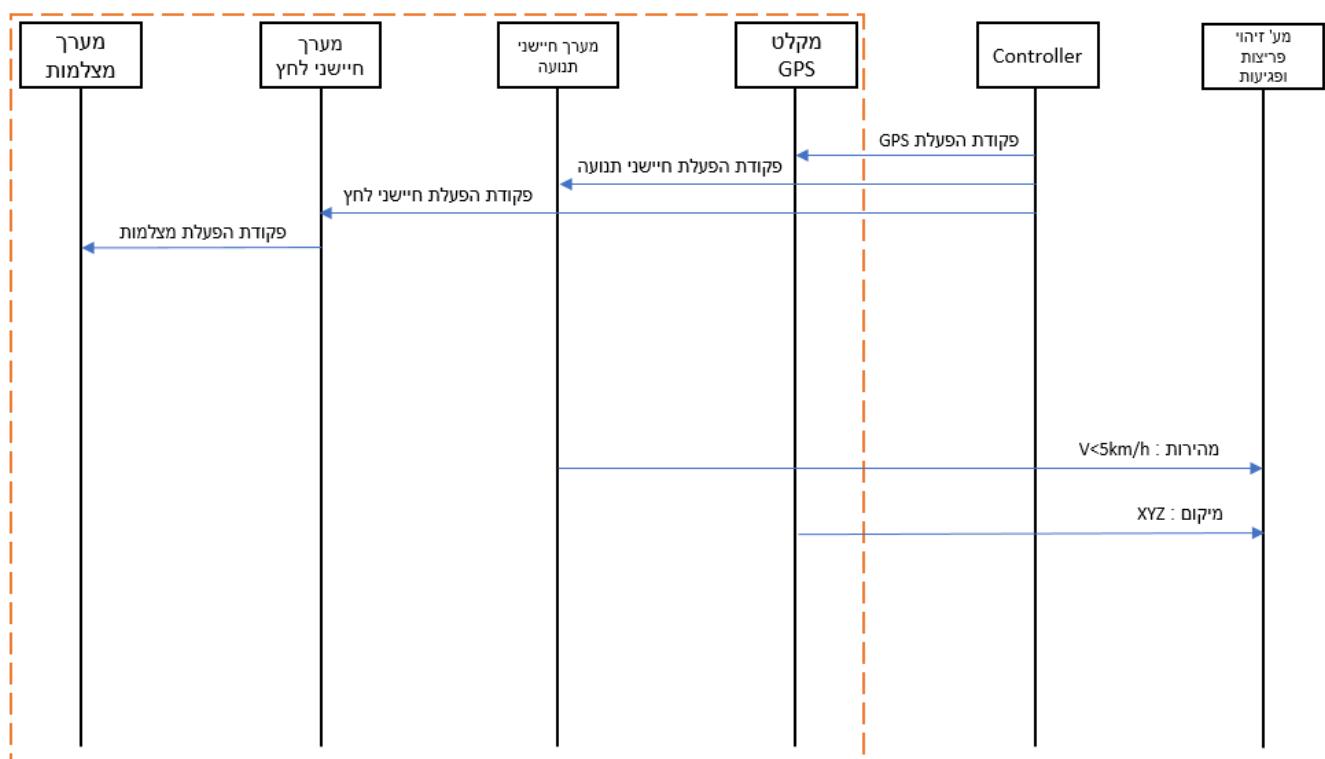
התדרדרות של הרכיב מהנניה בירידה (מצב N) – מופע 1 – קופסה שחורה:



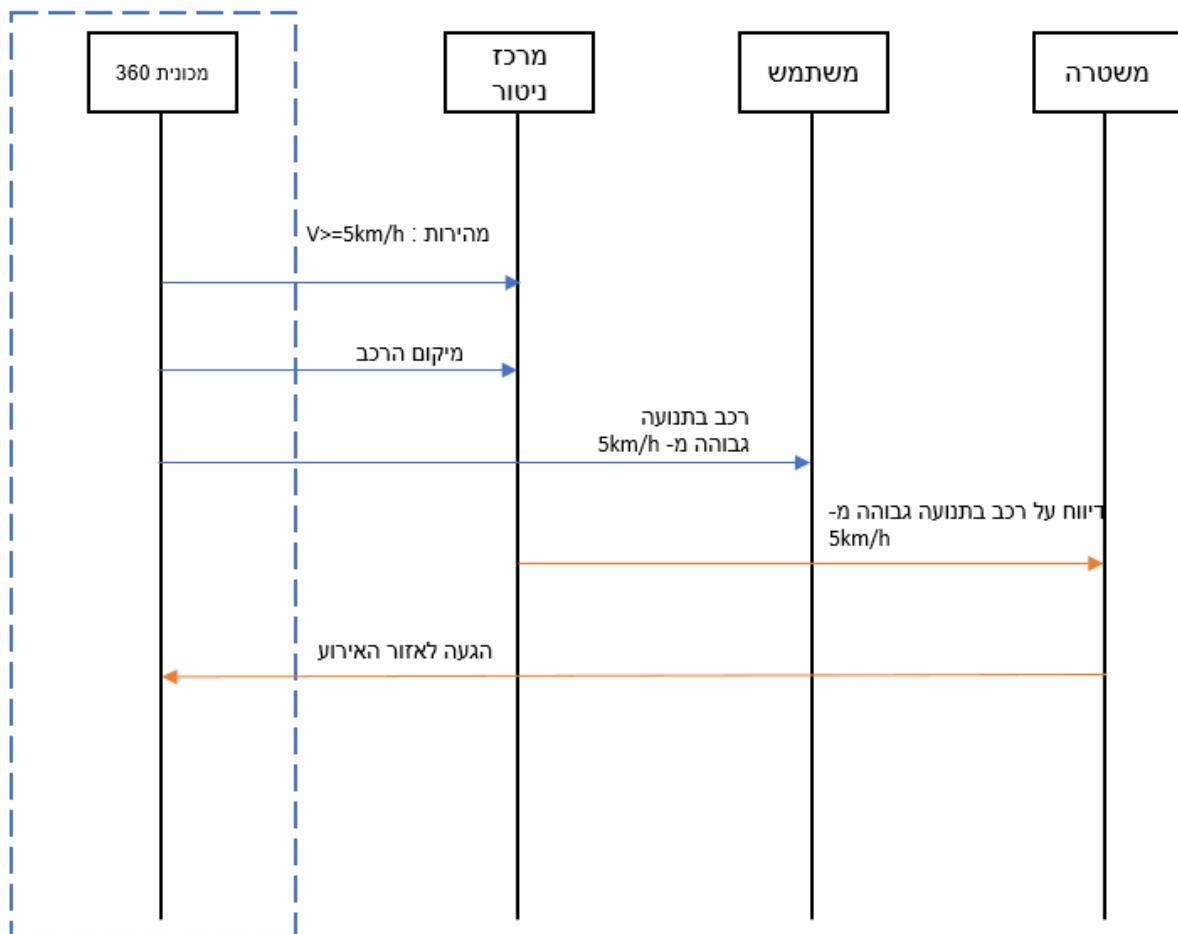
התדרדרות של הרכיב מחנית בירידה (מצב A) – מופע 1 – קופסה לבנה (שכבה 1):



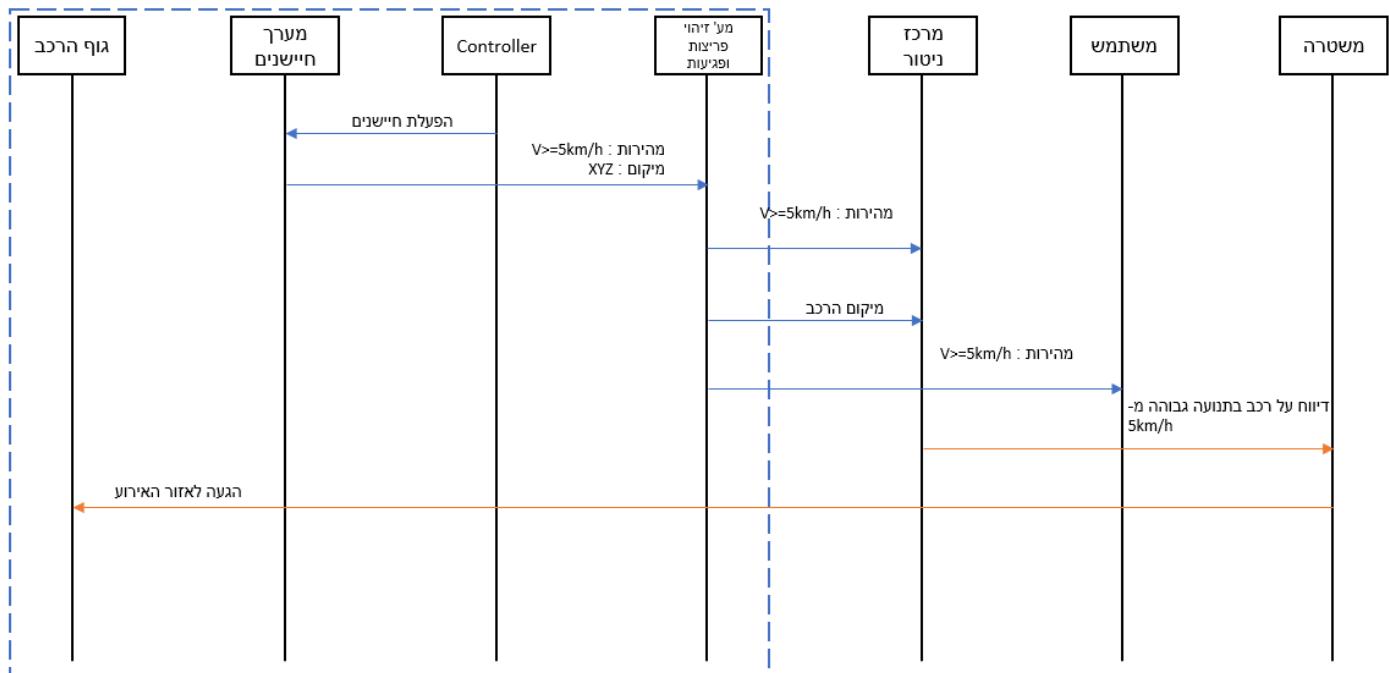
**התדרדרות של הרכיב מחנית בירידה (מצב A) – מופע 1 – קופסה לבנה (שכבה 2):**



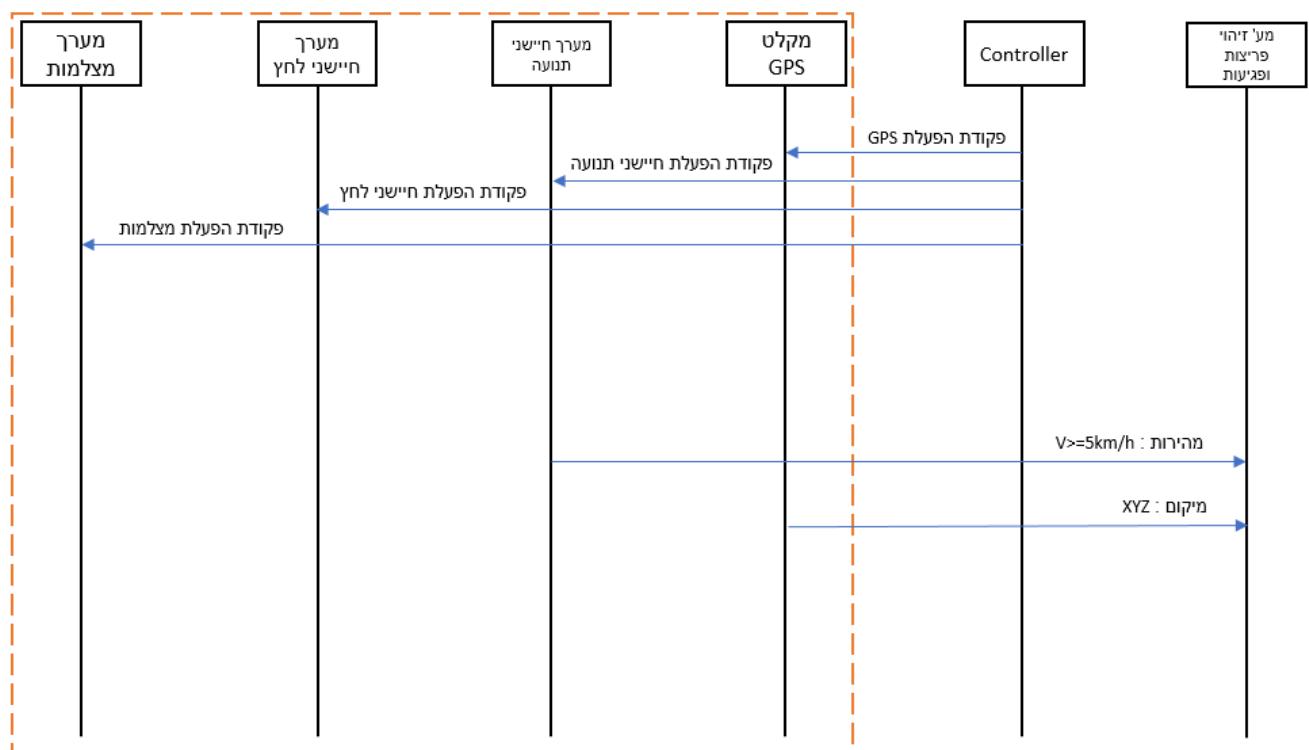
**התדרדרות של הרכב מחנית בירידה (מצב A) – מופע 2 – קופסה שחורה:**



**התדרדרות של הרכב מחנית בירידה (מצב A) – מופע 2 – קופסה לבנה (שכבה 1):**



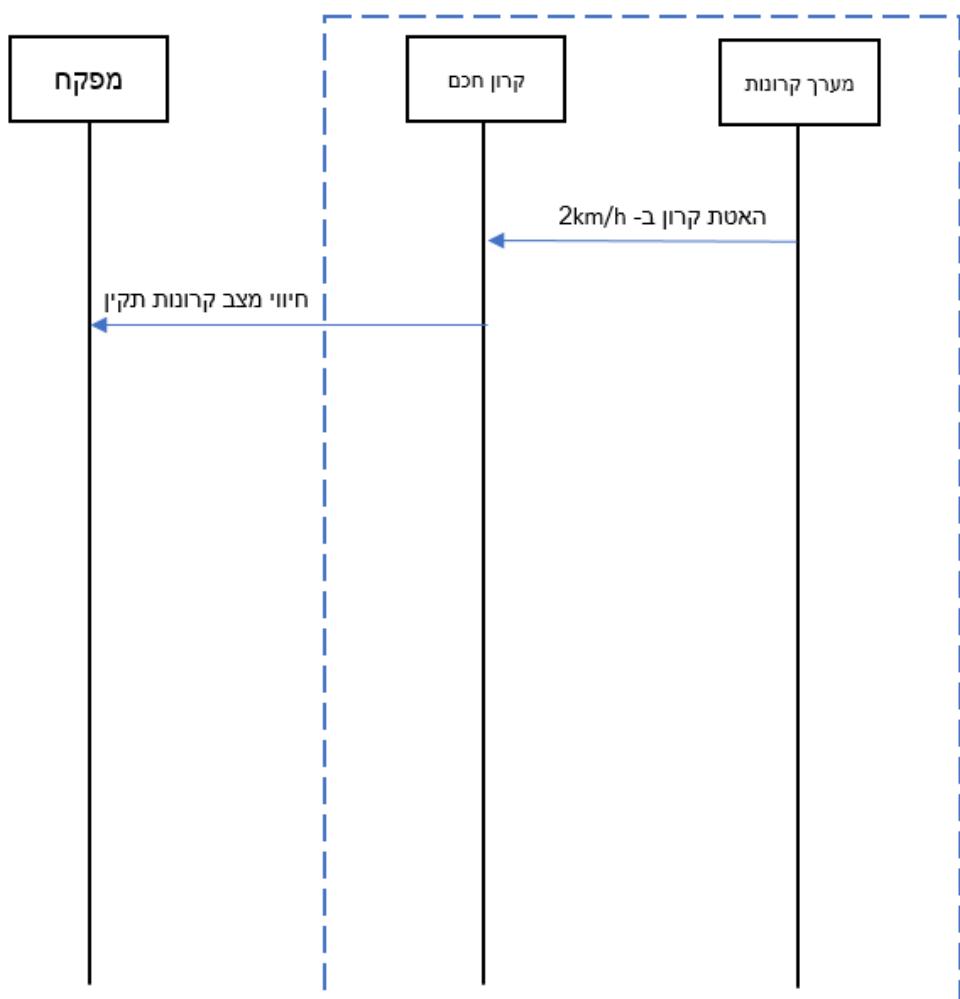
התדרדרות של הרכיב מחניתה בירידה (מצב A) – מופע 2 – קופסה לבנה (שכבה 2):



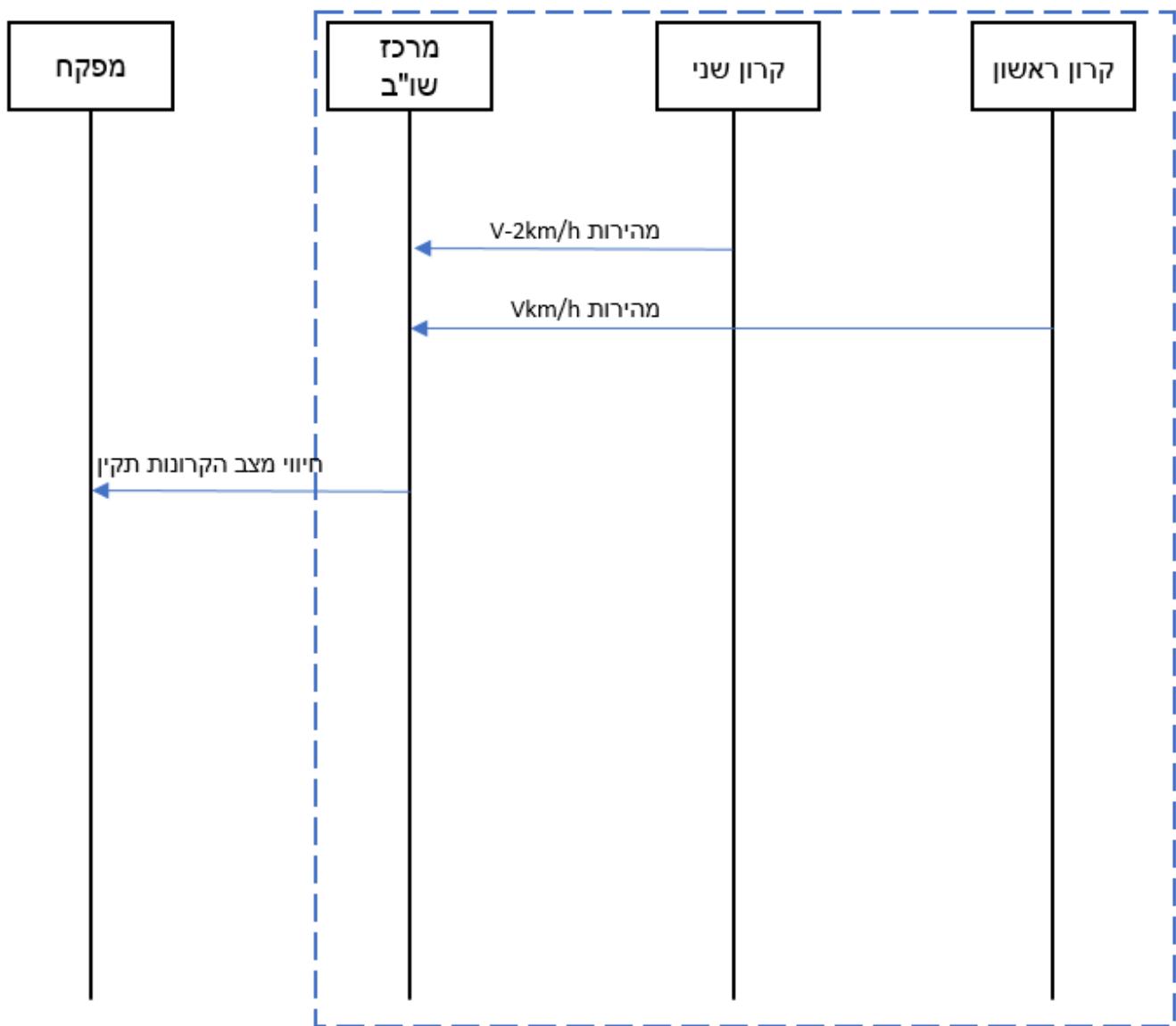
**ב-2 - ניהול מערך קرونות חכם (מתיחס לסעיף 3.2.2):**

**ב-2.1 - האטת מהירות:**

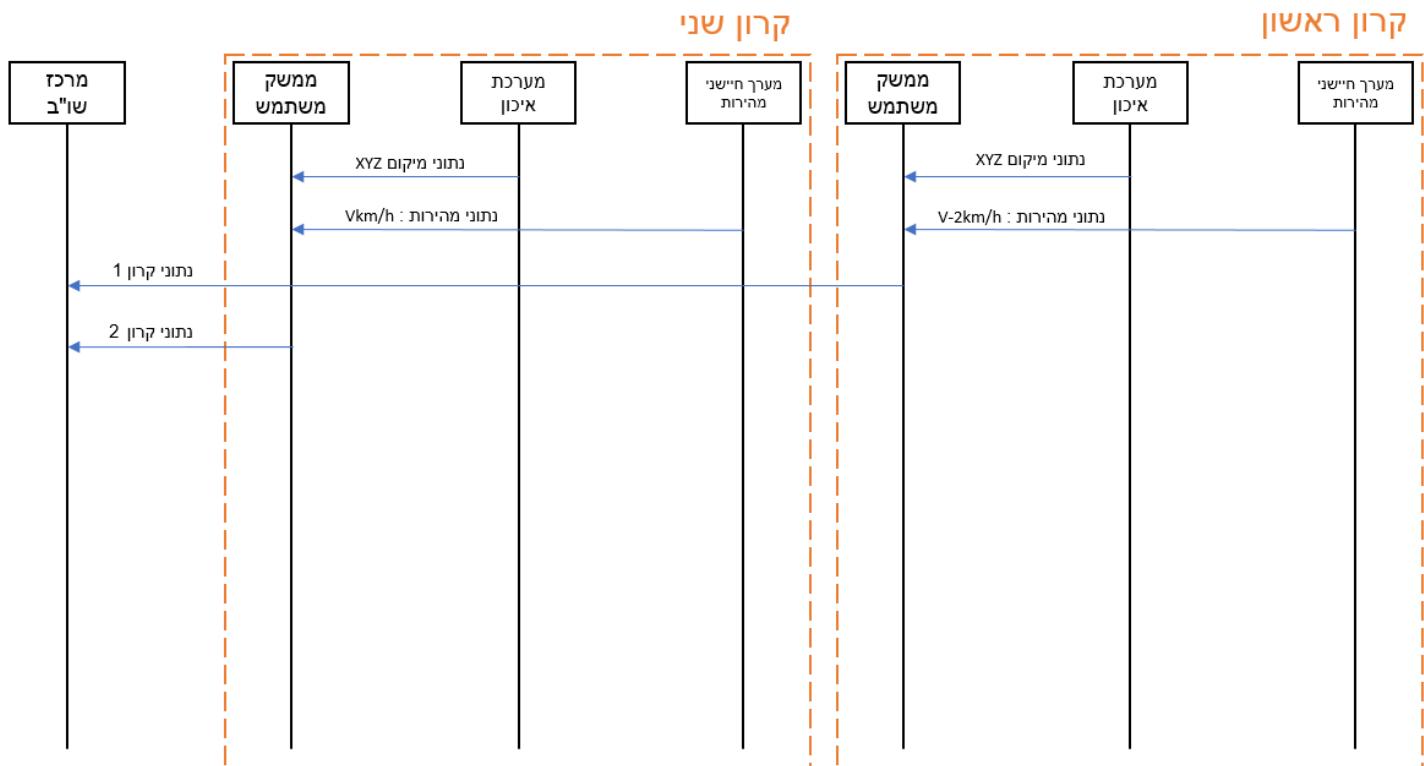
**האטת מהירות – מופע 1 – קופסה שחורה:**



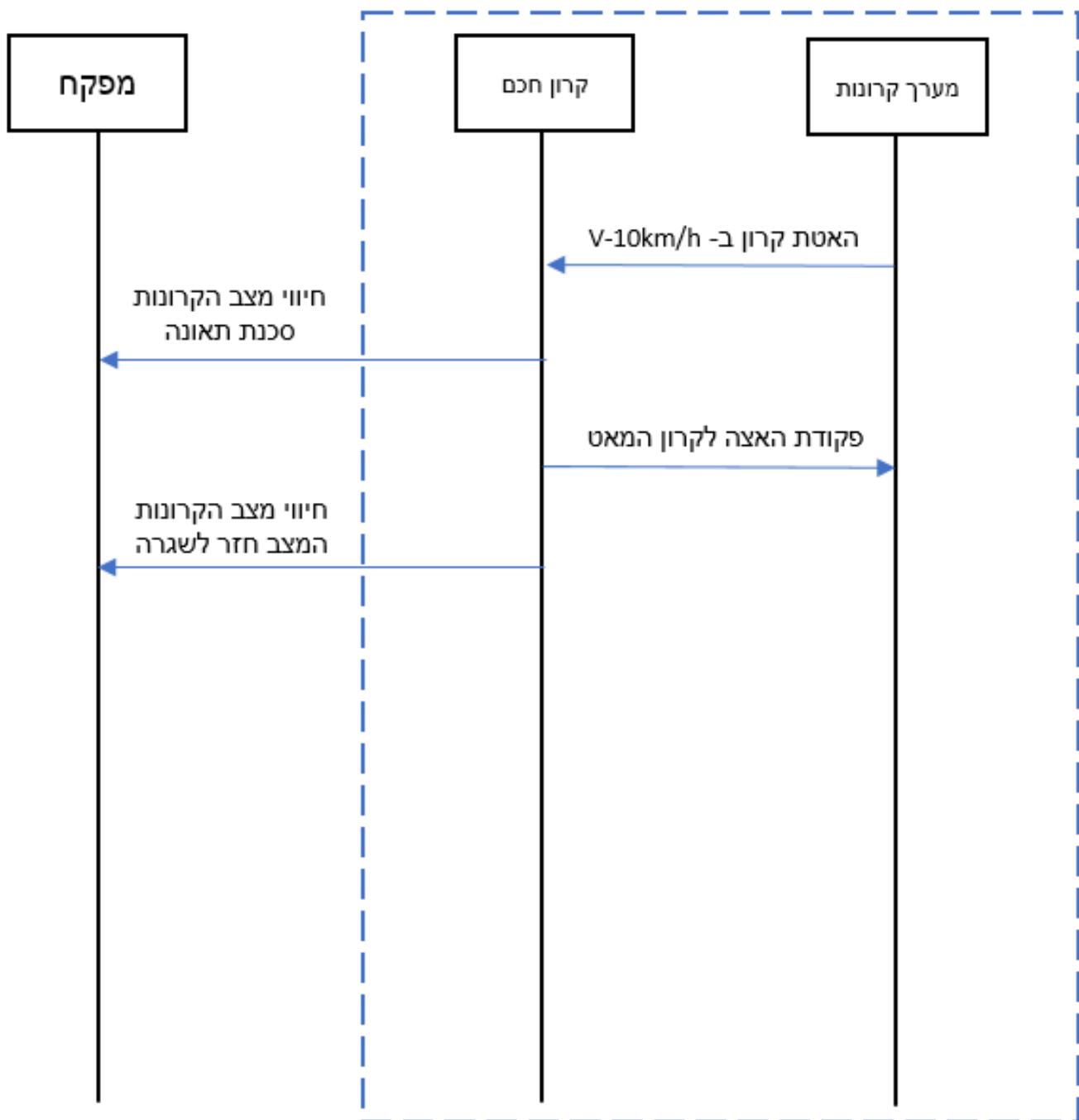
האטת מהירות – מופע 1 – קופסה לבנה (שכבה 1):



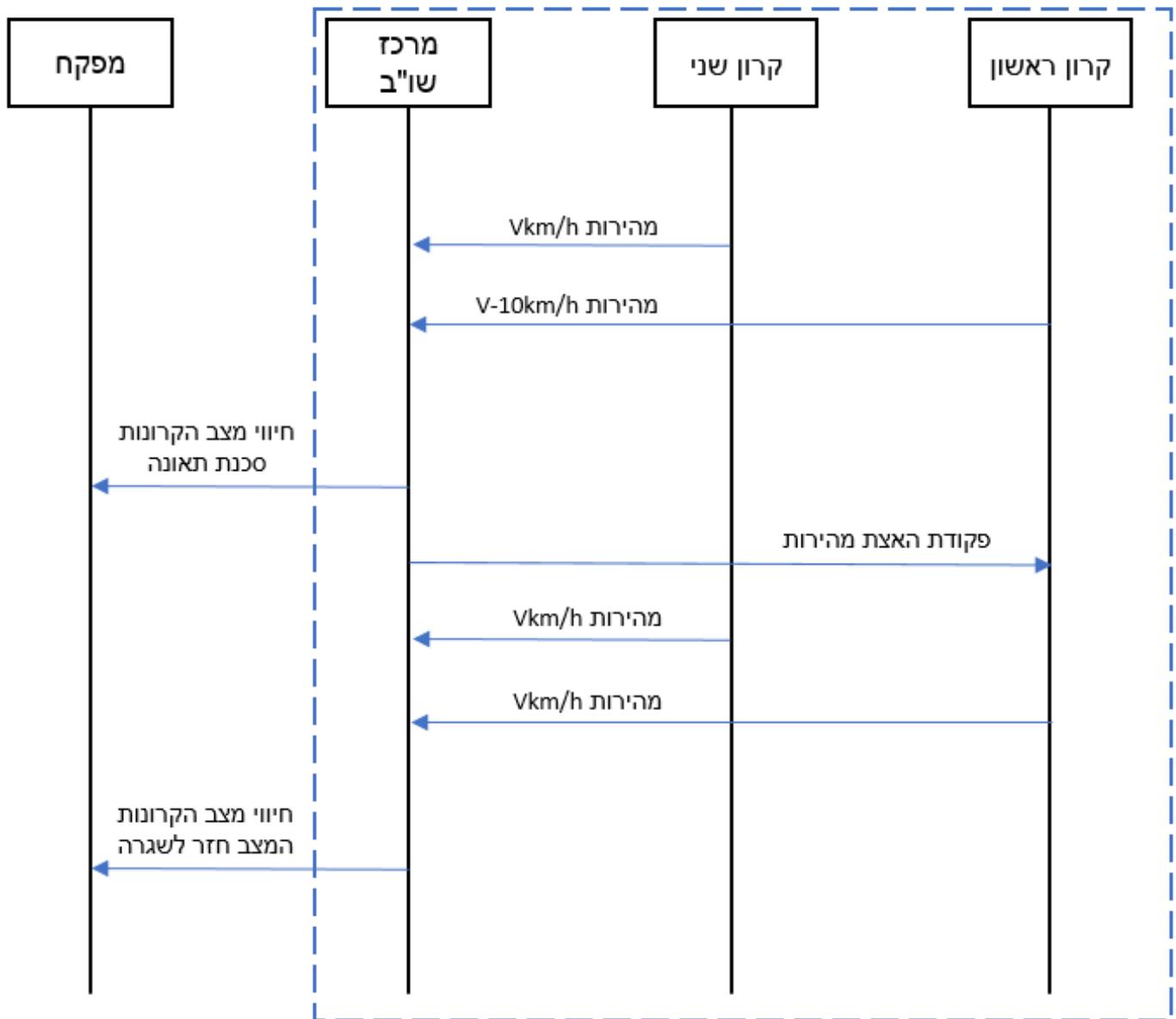
האטת מהירות – מופע 1 – קופסה לבנה (שכבה 2):



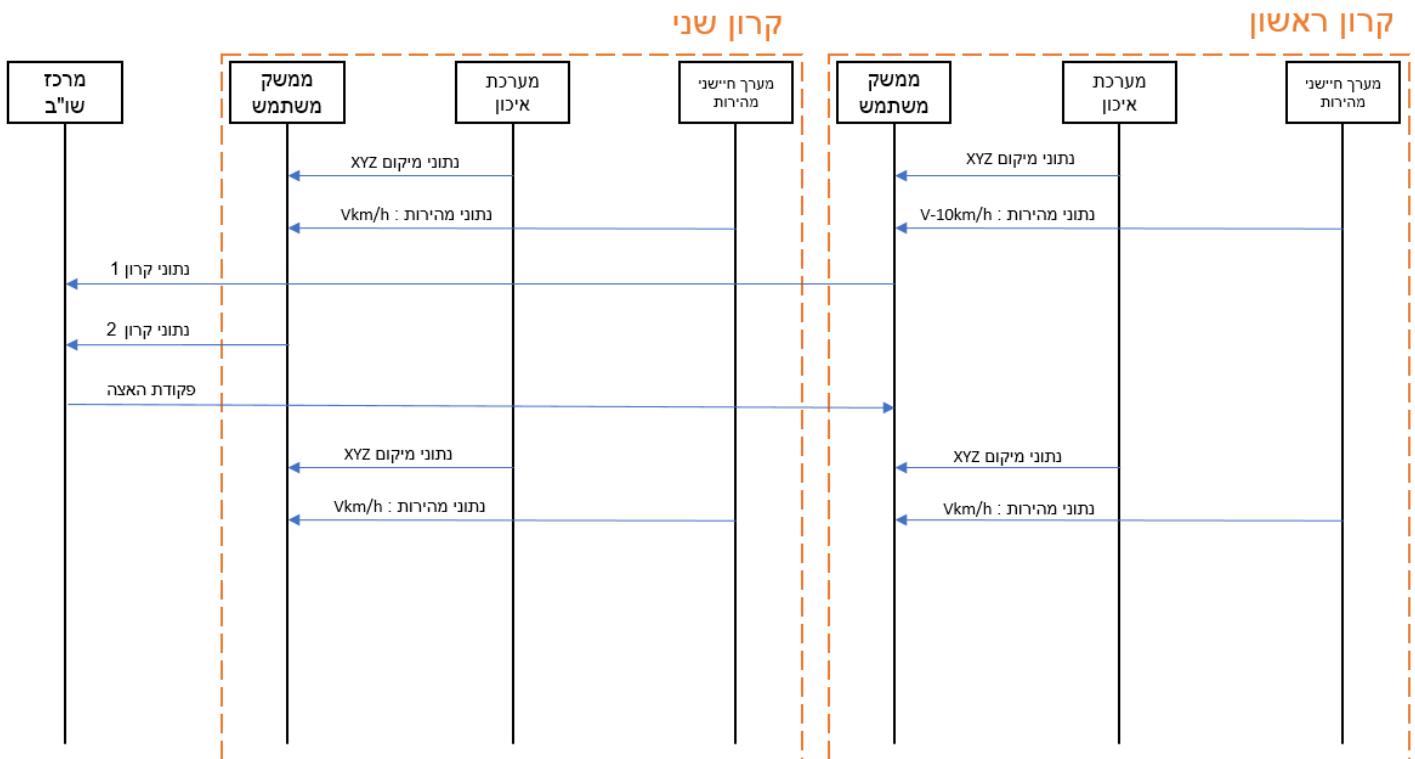
האטת מהירות – מופע 2 – קופסה שחורה:



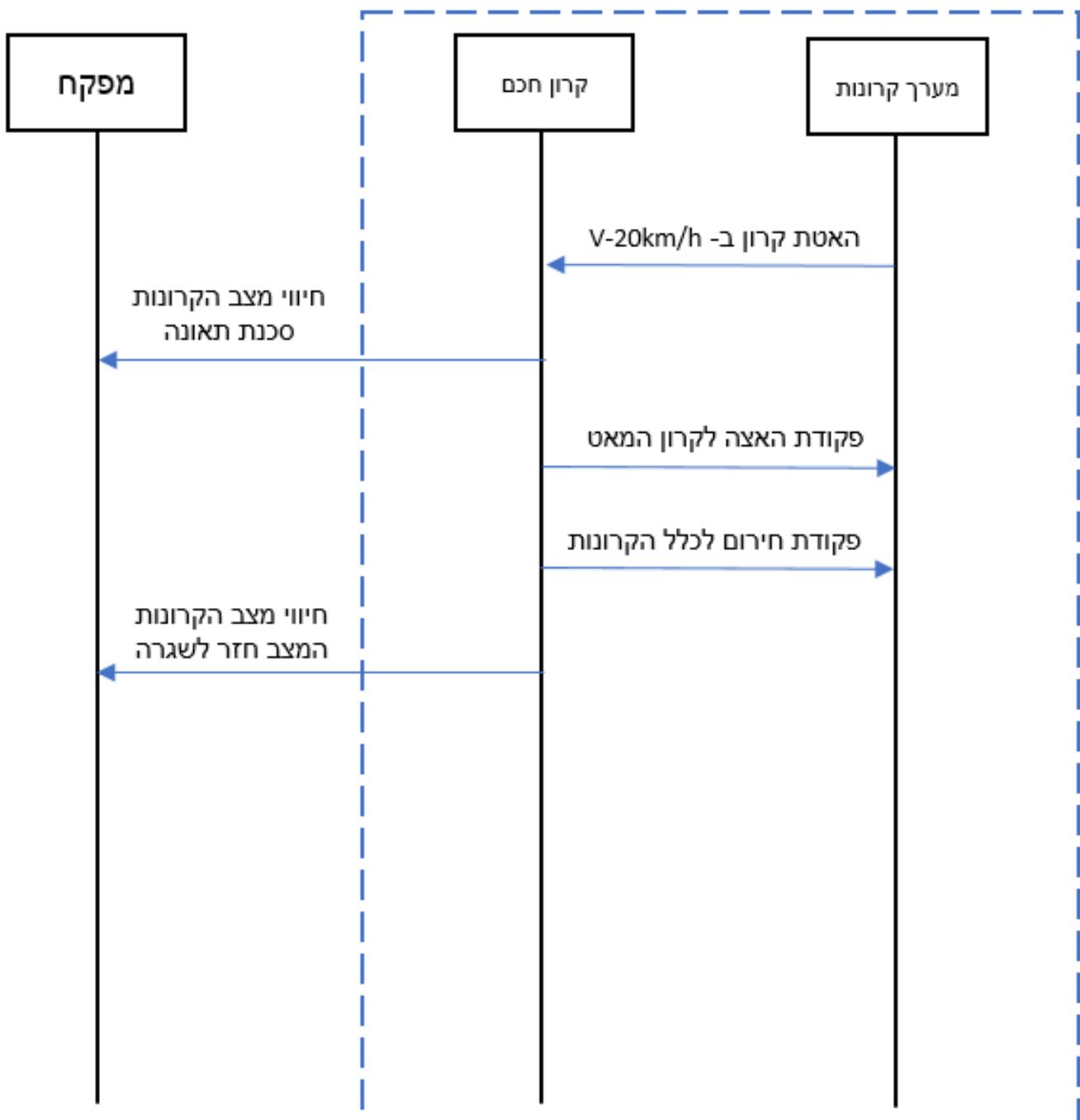
האטת מהירות – מופע 2 – קופסה לבנה (שכבה 1):



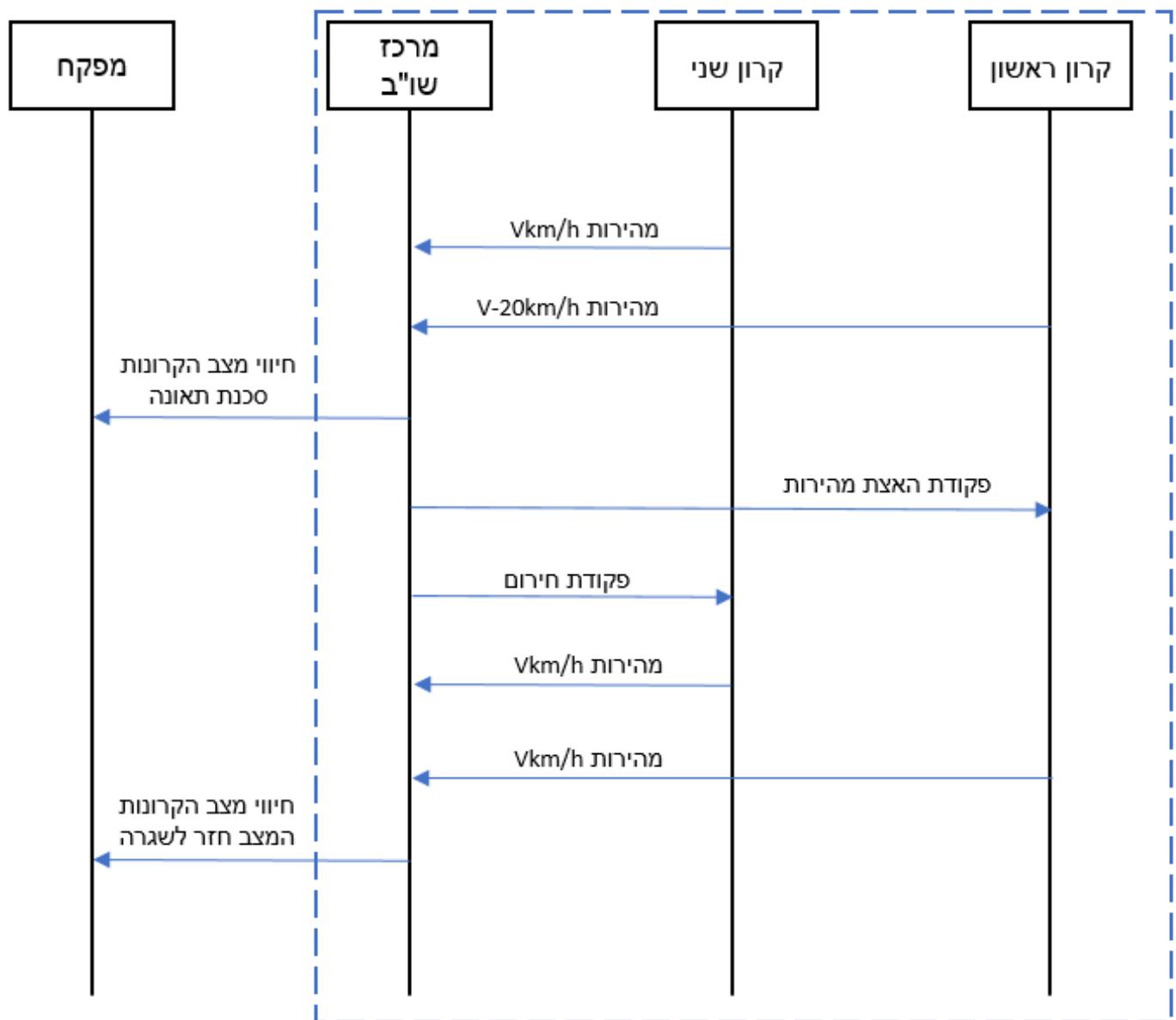
האטת מהירות – מופע 2 – קופסה לבנה (שכבה 2):



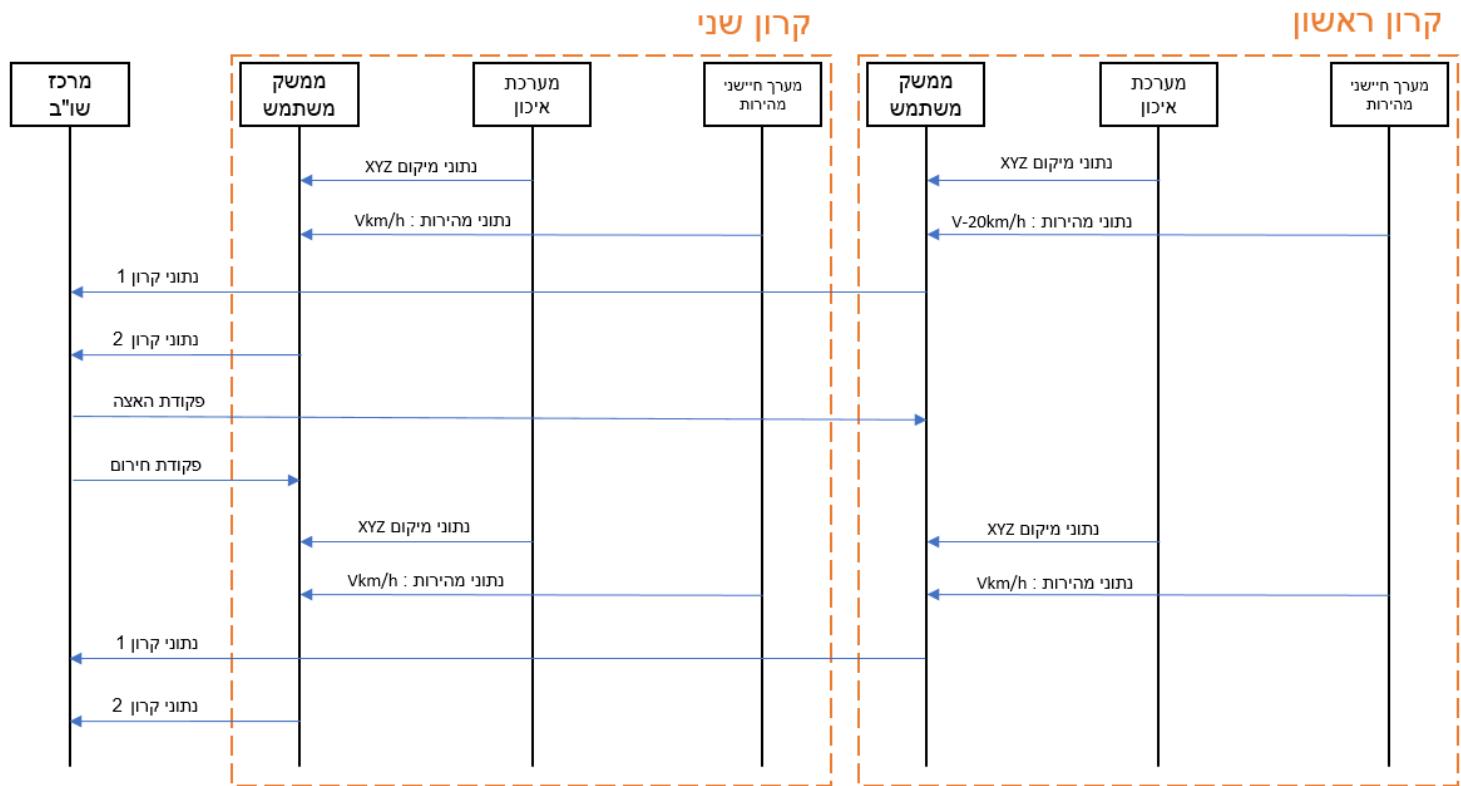
האטת מהירות – מופע 3 – קופסה שחורה:



האטת מהירות – מופע 3 – קופסה לבנה (שכבה 1):

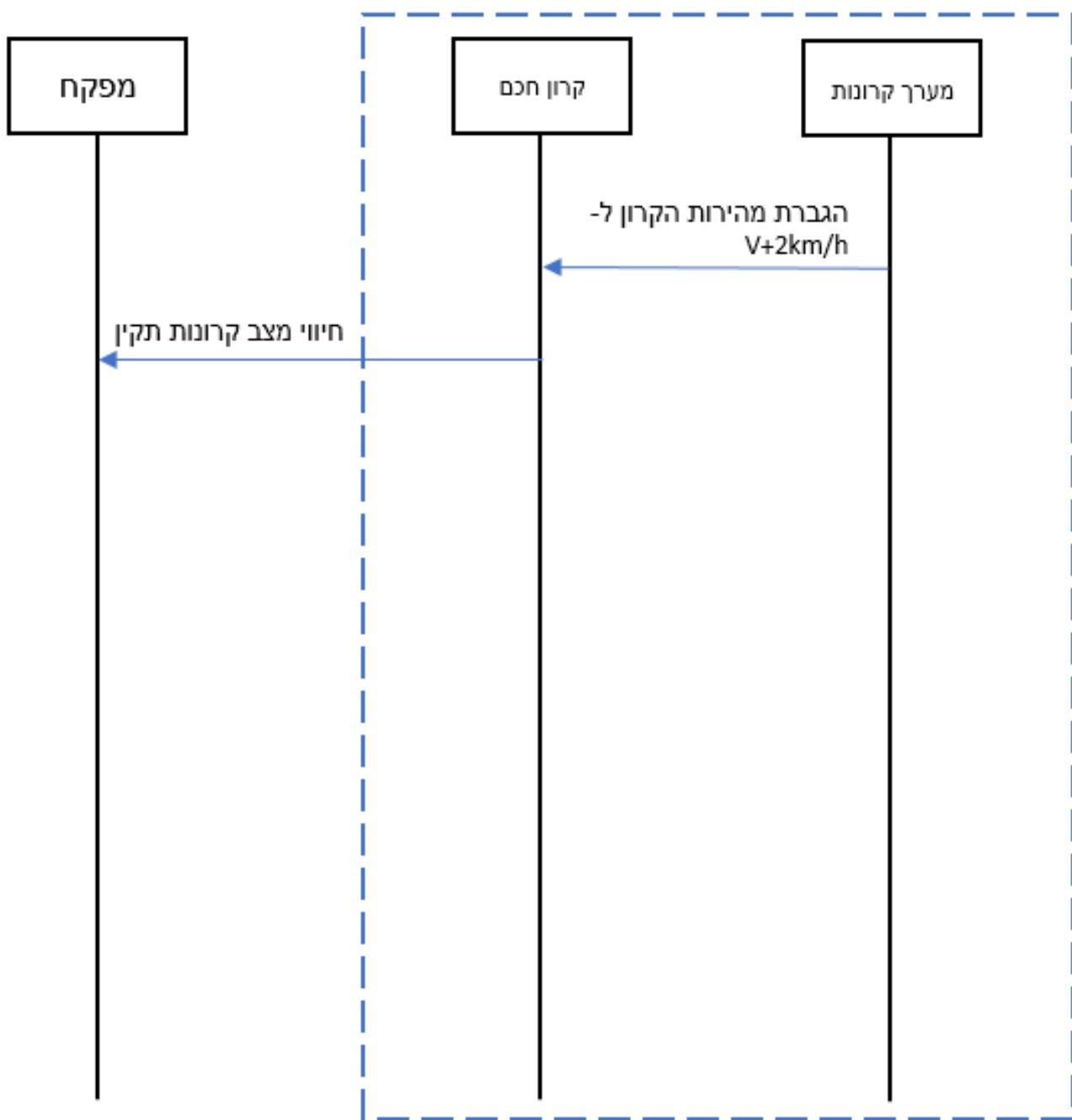


האטת מהירות – מופע 3 – קופסה לבנה (שכבה 2):

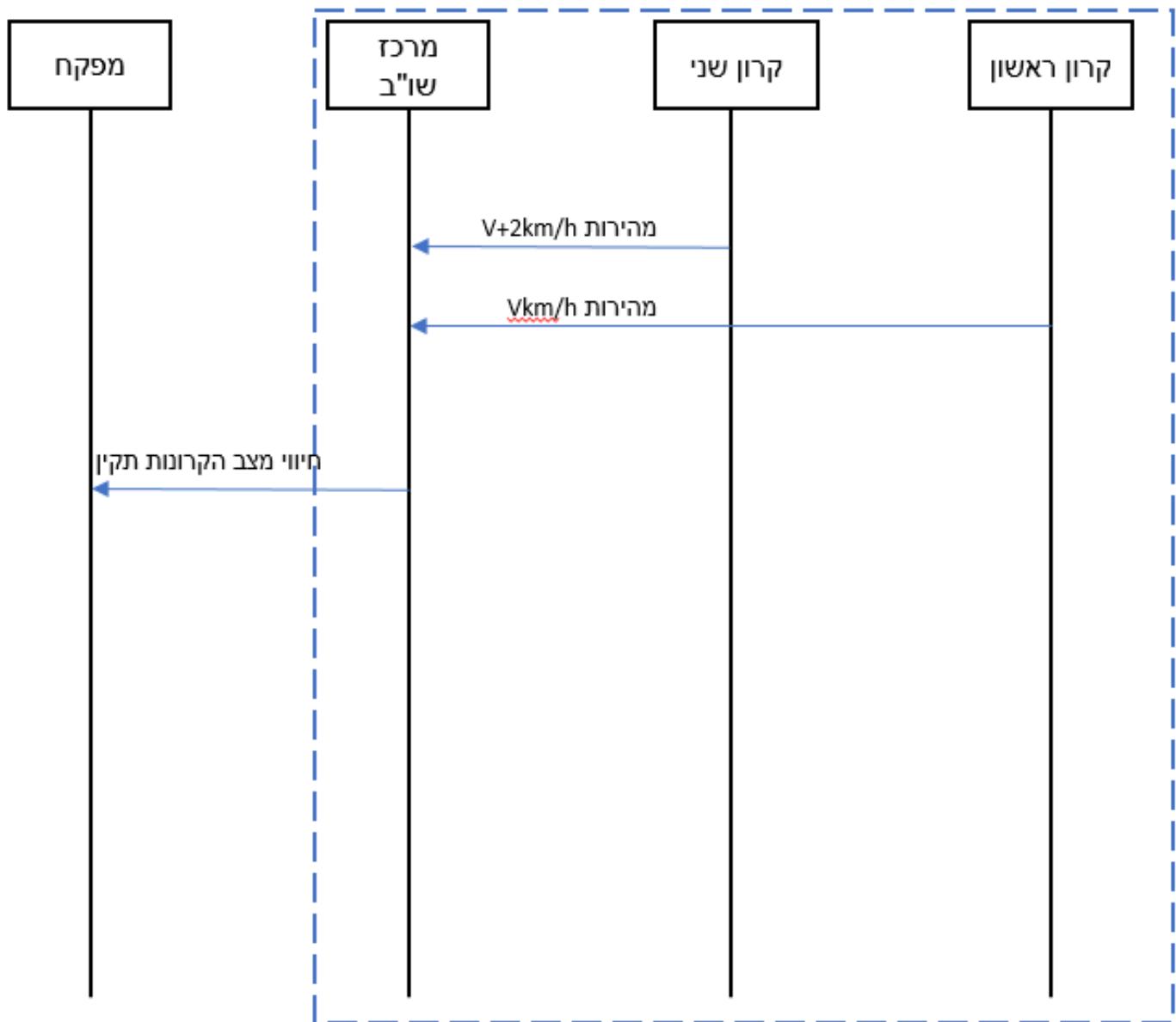


ב-2.2 – הגברת מהירות:

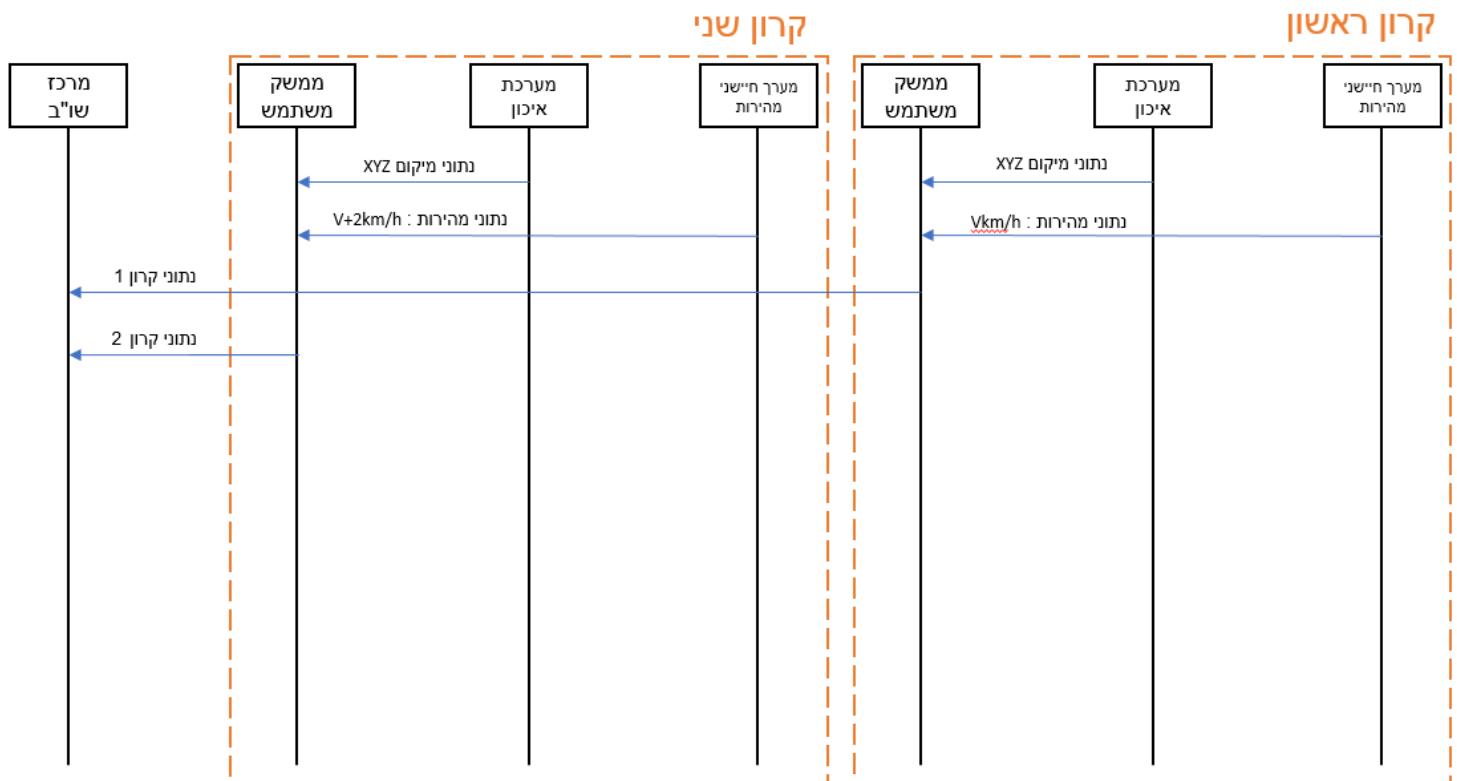
הגברת מהירות – מופע 1 – קופסה שחורה:



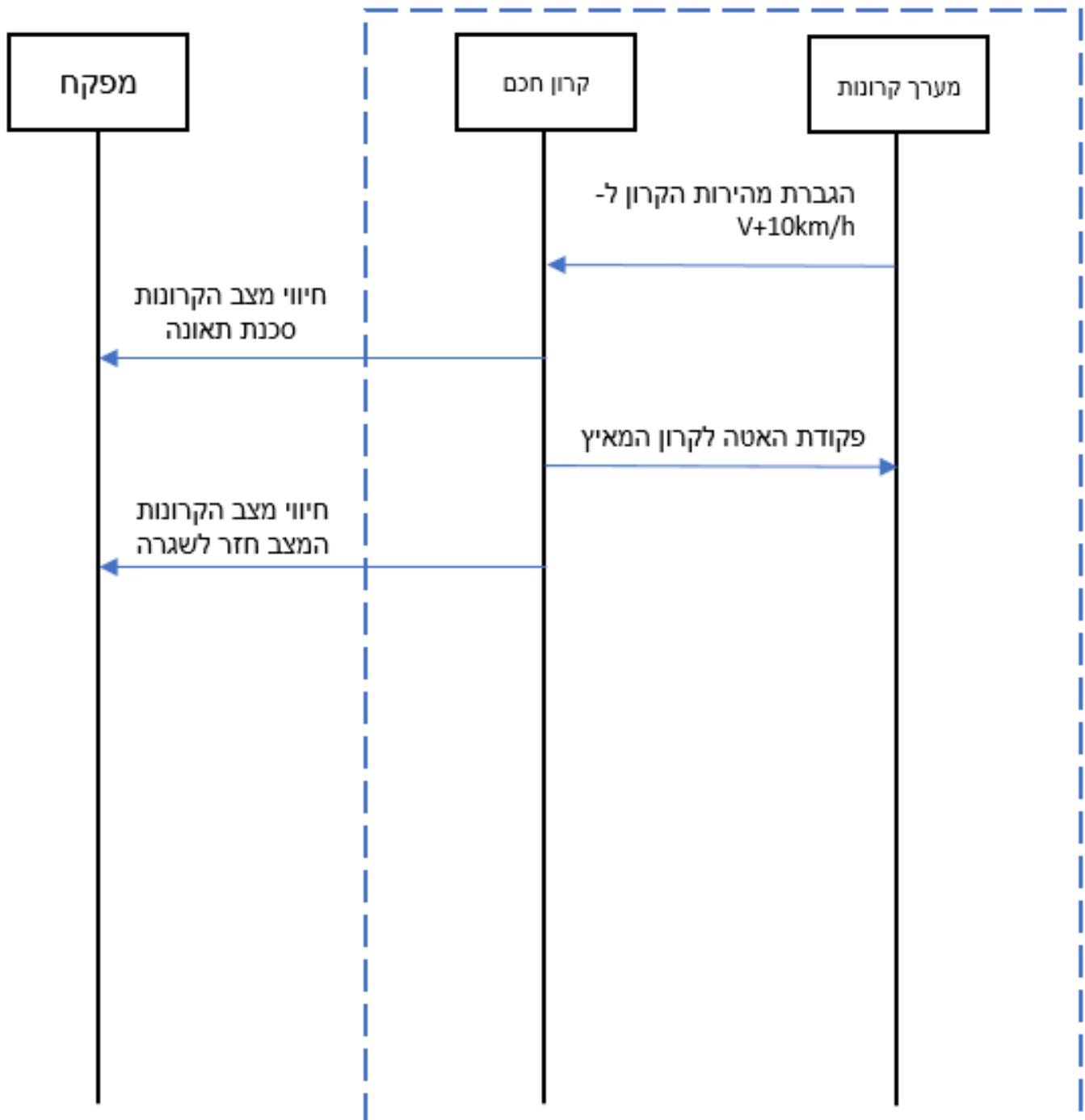
הגברת מהירות – מופע 1 – קופסה לבנה (שכבה 1):



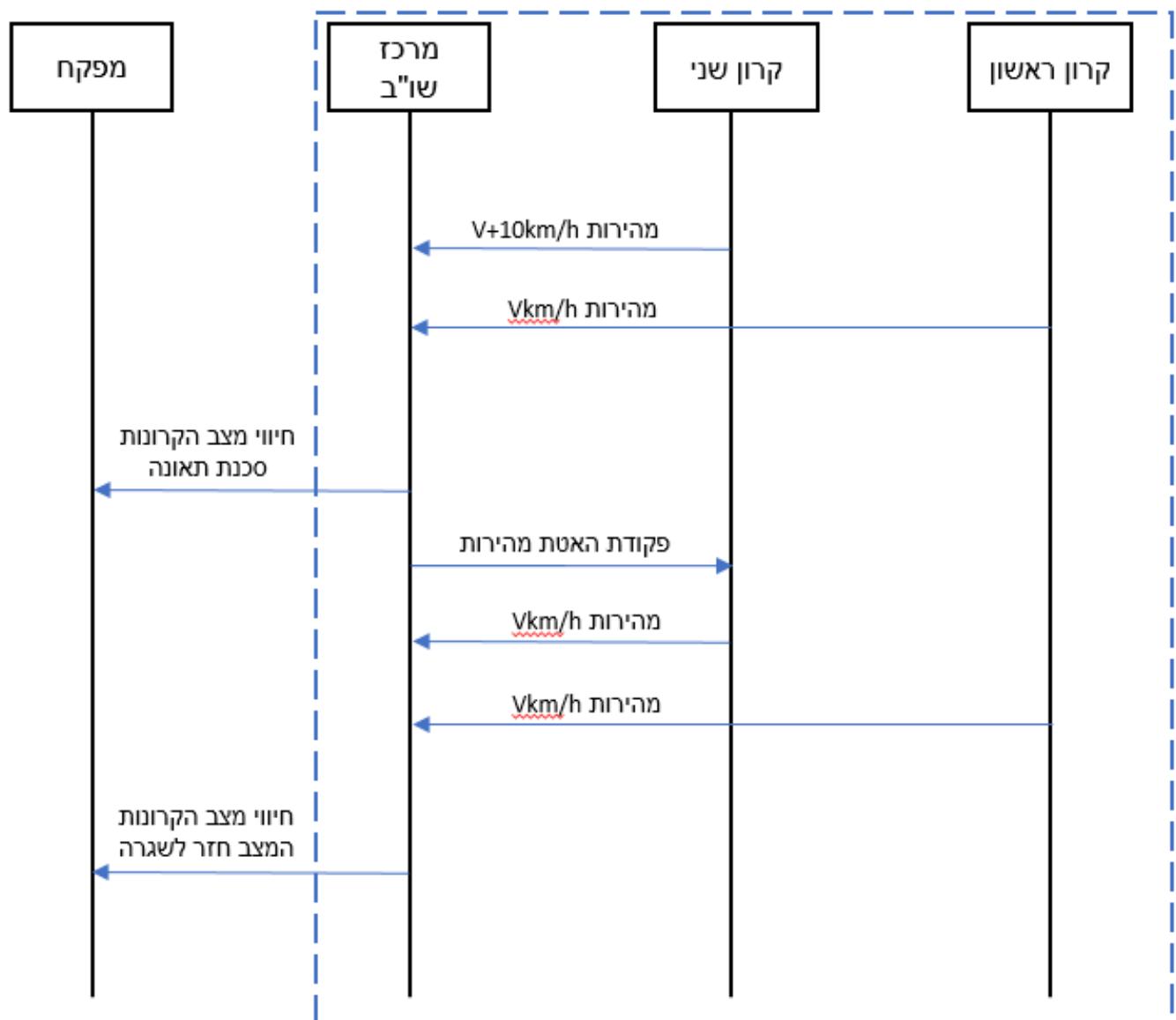
הגברת מהירות – מופע 1 – קופסה לבנה (שכבה 2):



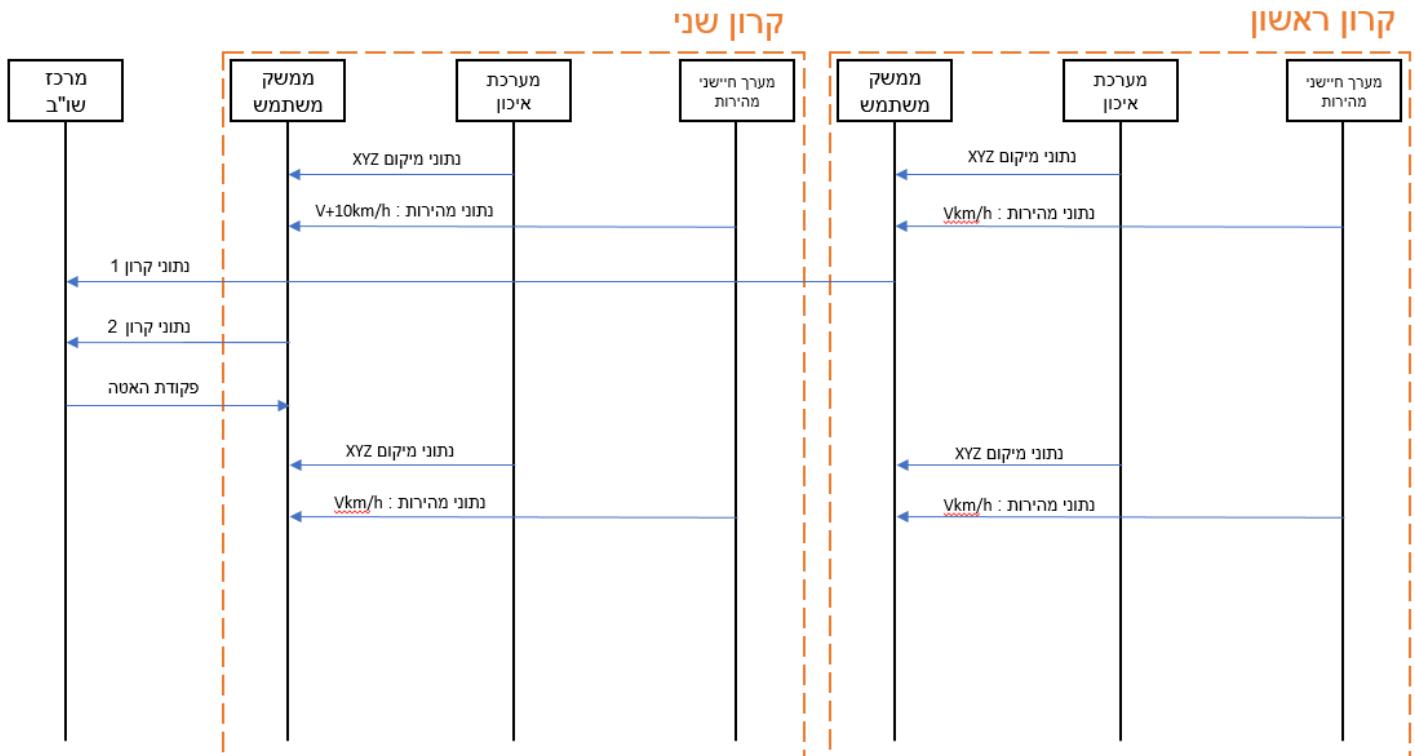
הגברת מהירות – מופע 2 – קופסה שחורה:



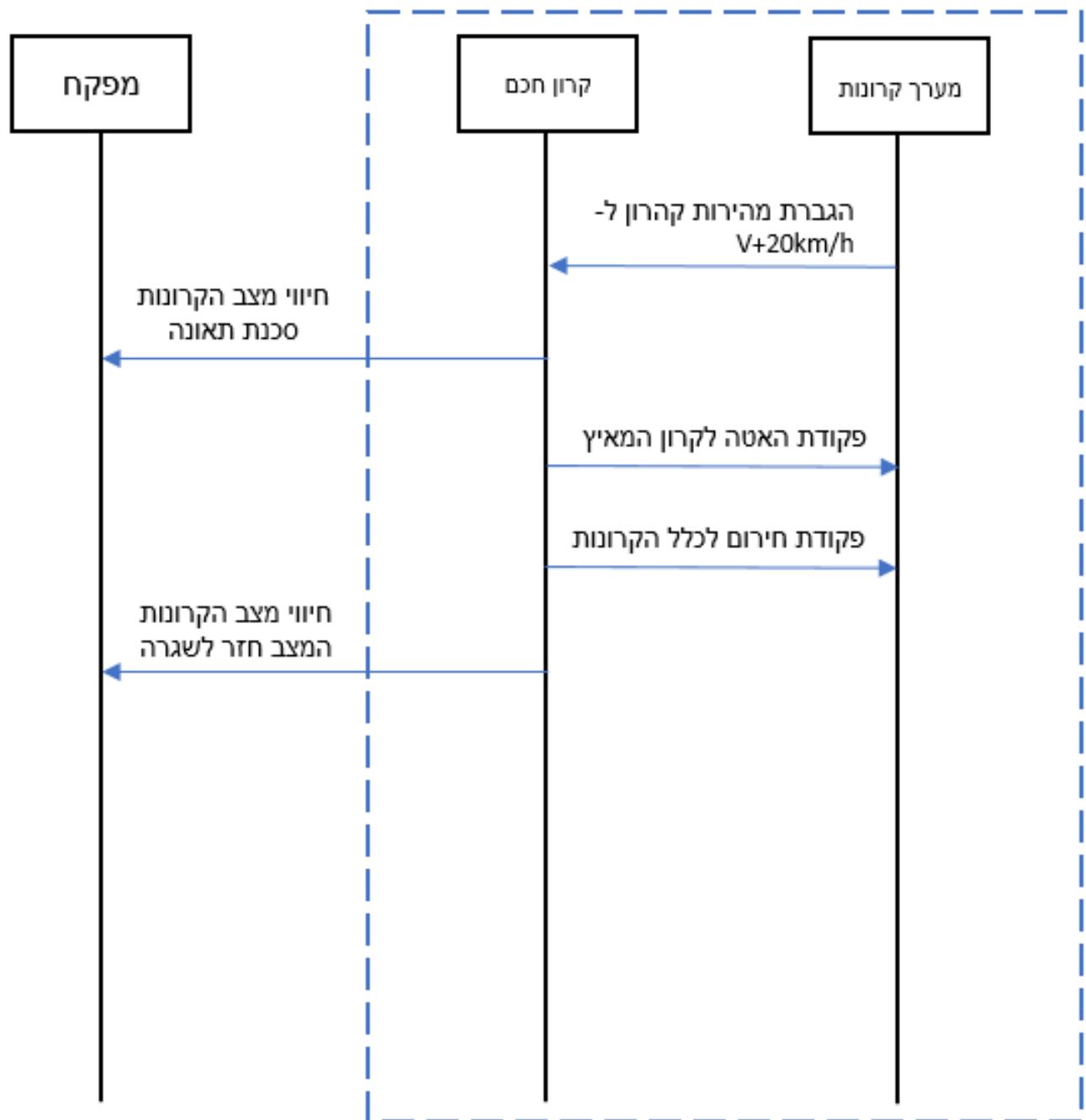
הגברת מהירות – מופע 2 – קופסה לבנה (שכבה 1):



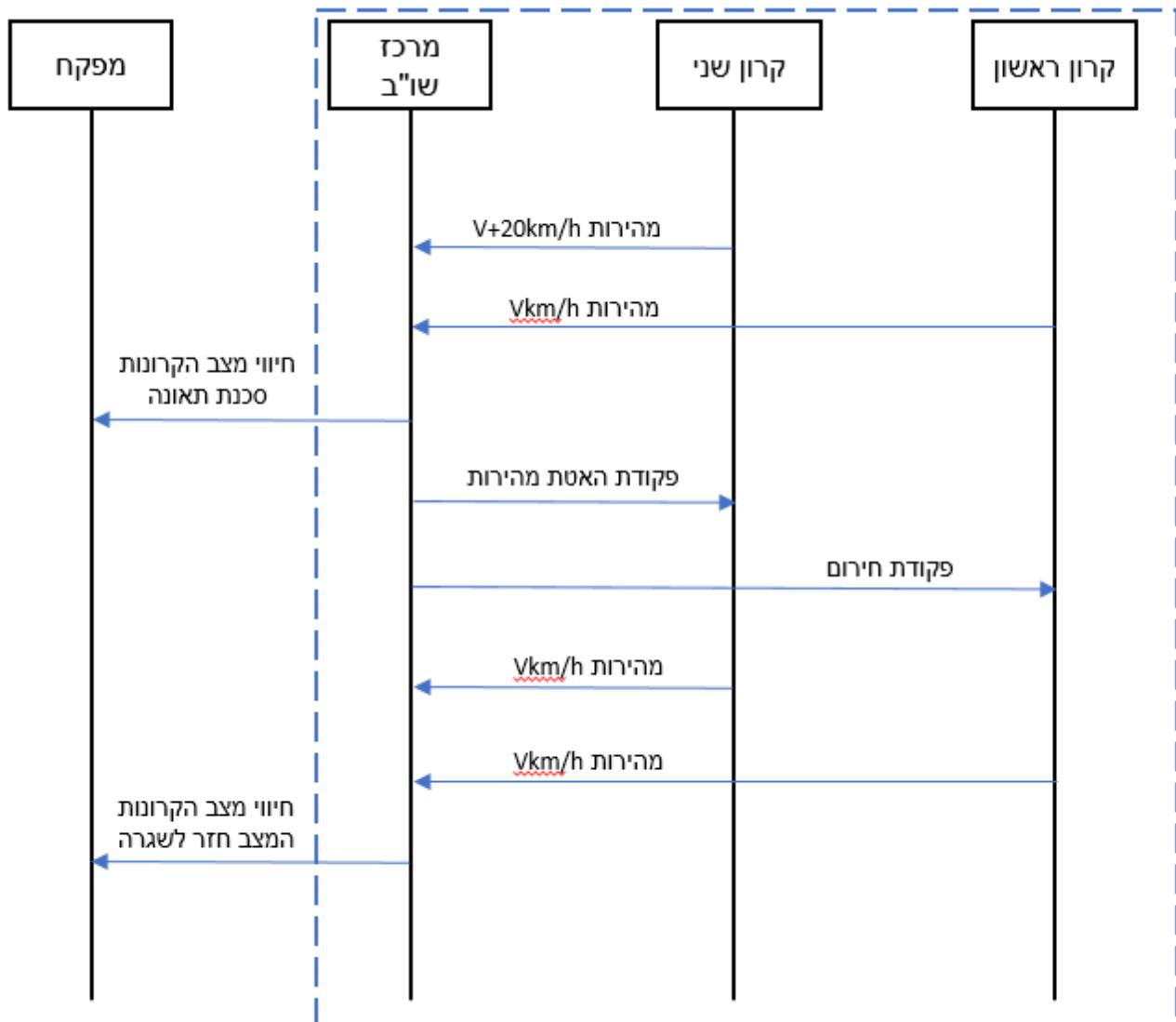
הגברת מהירות – מופע 2 – קופסה לבנה (שכבה 2):



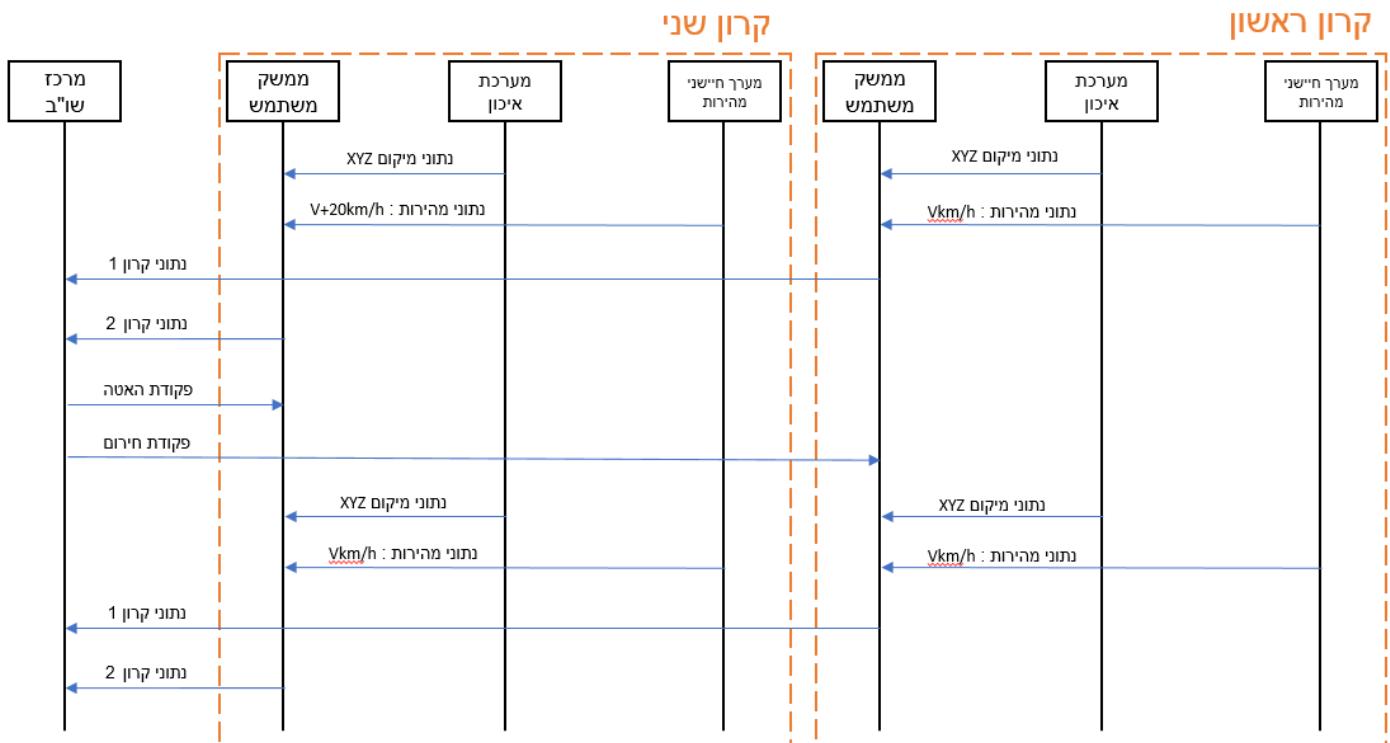
**הגברת מהירות – מופע 3 – קופסה שחורה:**



הגברת מהירות – מופע 3 – קופסה לבנה (שכבה 1):

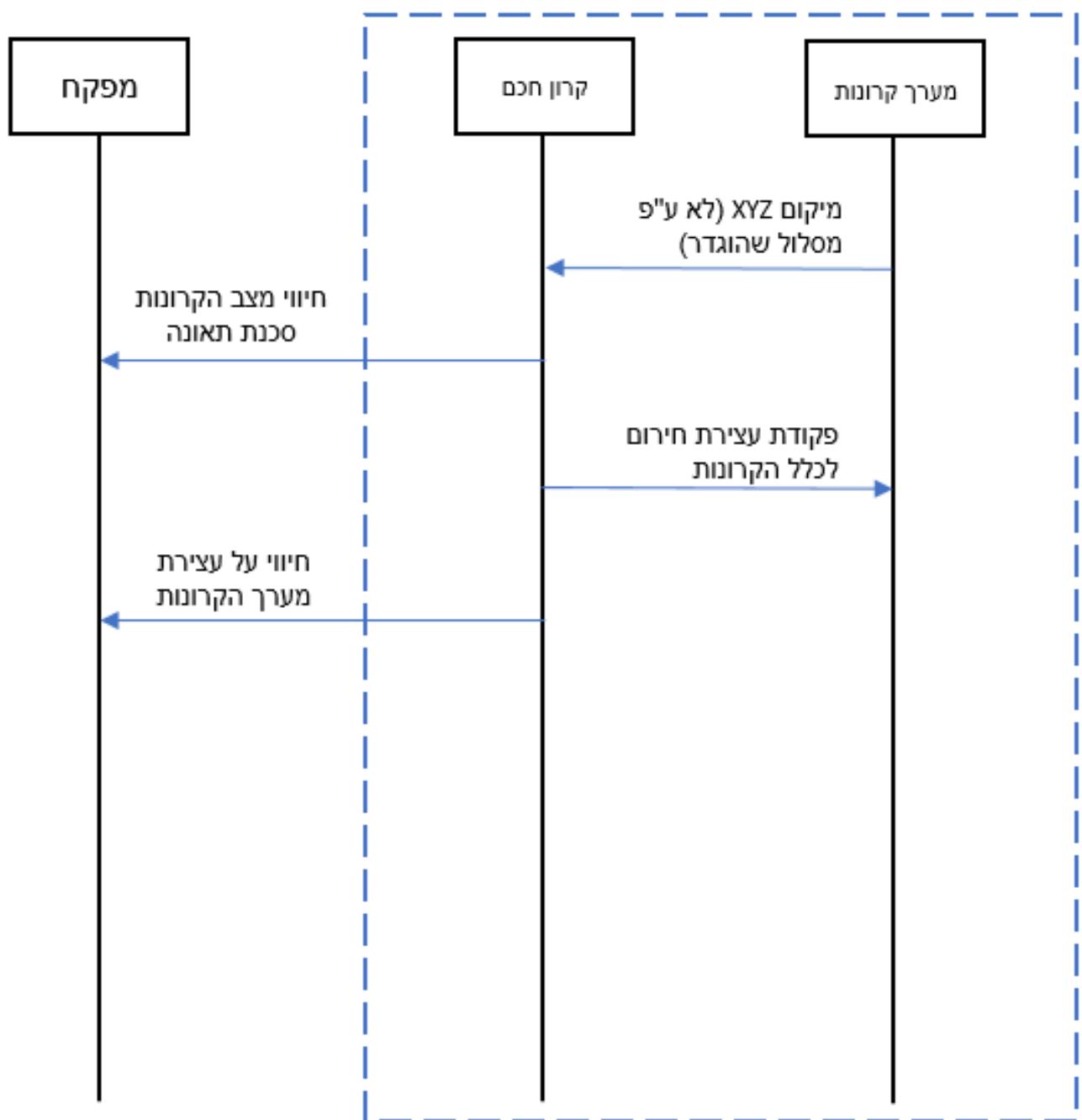


הגברת מהירות – מופע 3 – קופסה לבנה (שכבה 2):

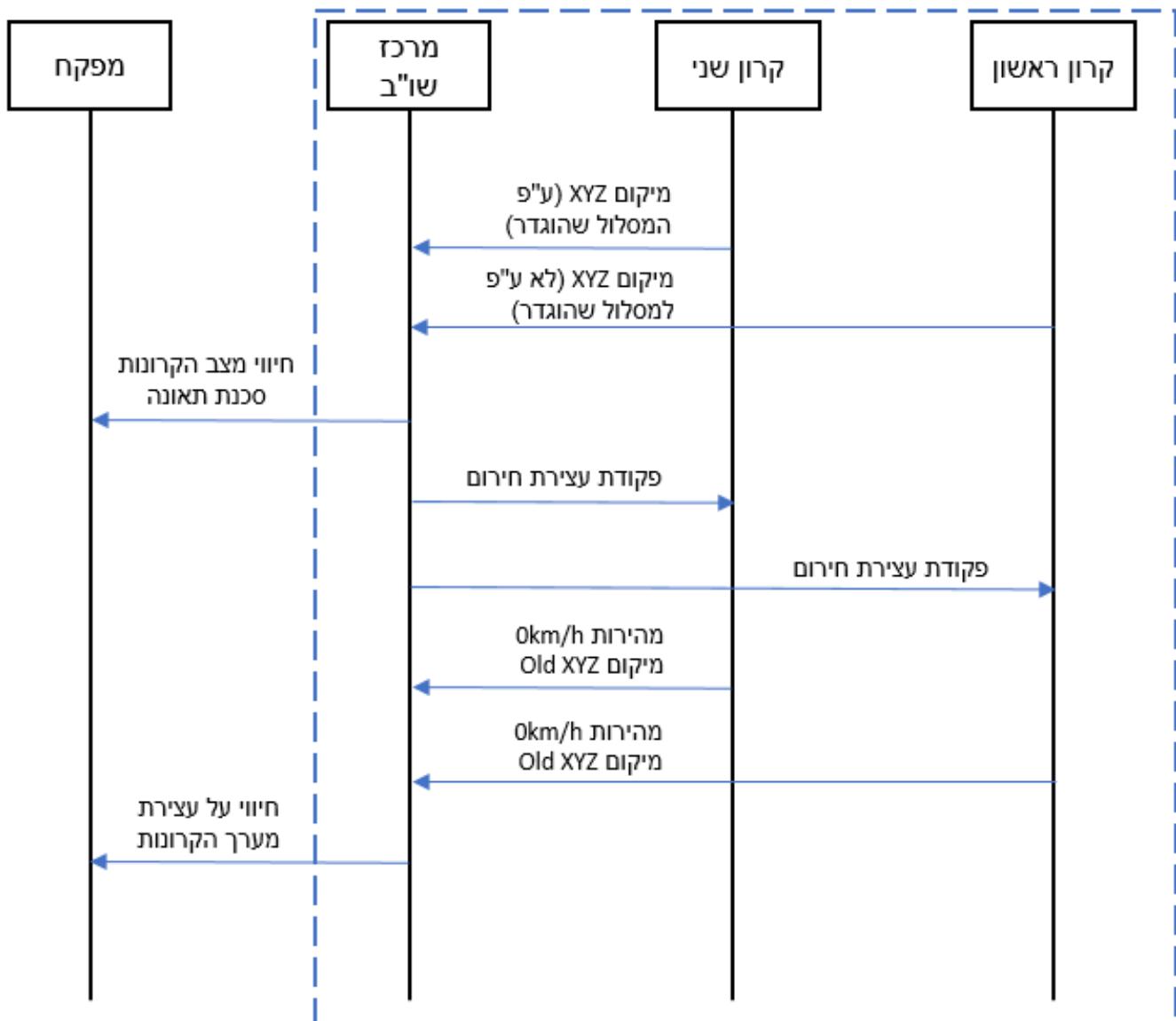


ב-2.3 - שינוי מיקום אחד הקرونנות (סטייה/תקיעה):

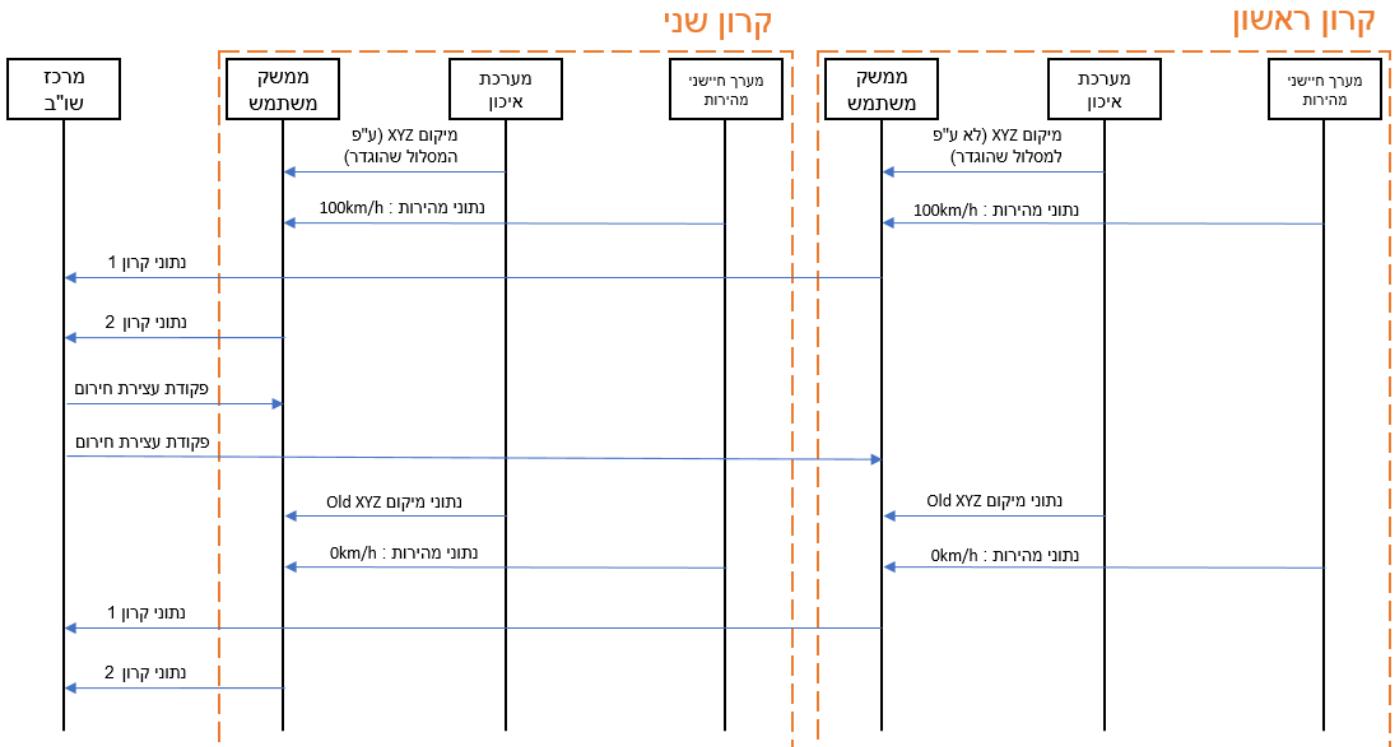
שינוי מיקום אחד הקرونנות (סטייה/תקיעה) – מופע 1 – קופסה שחורה:



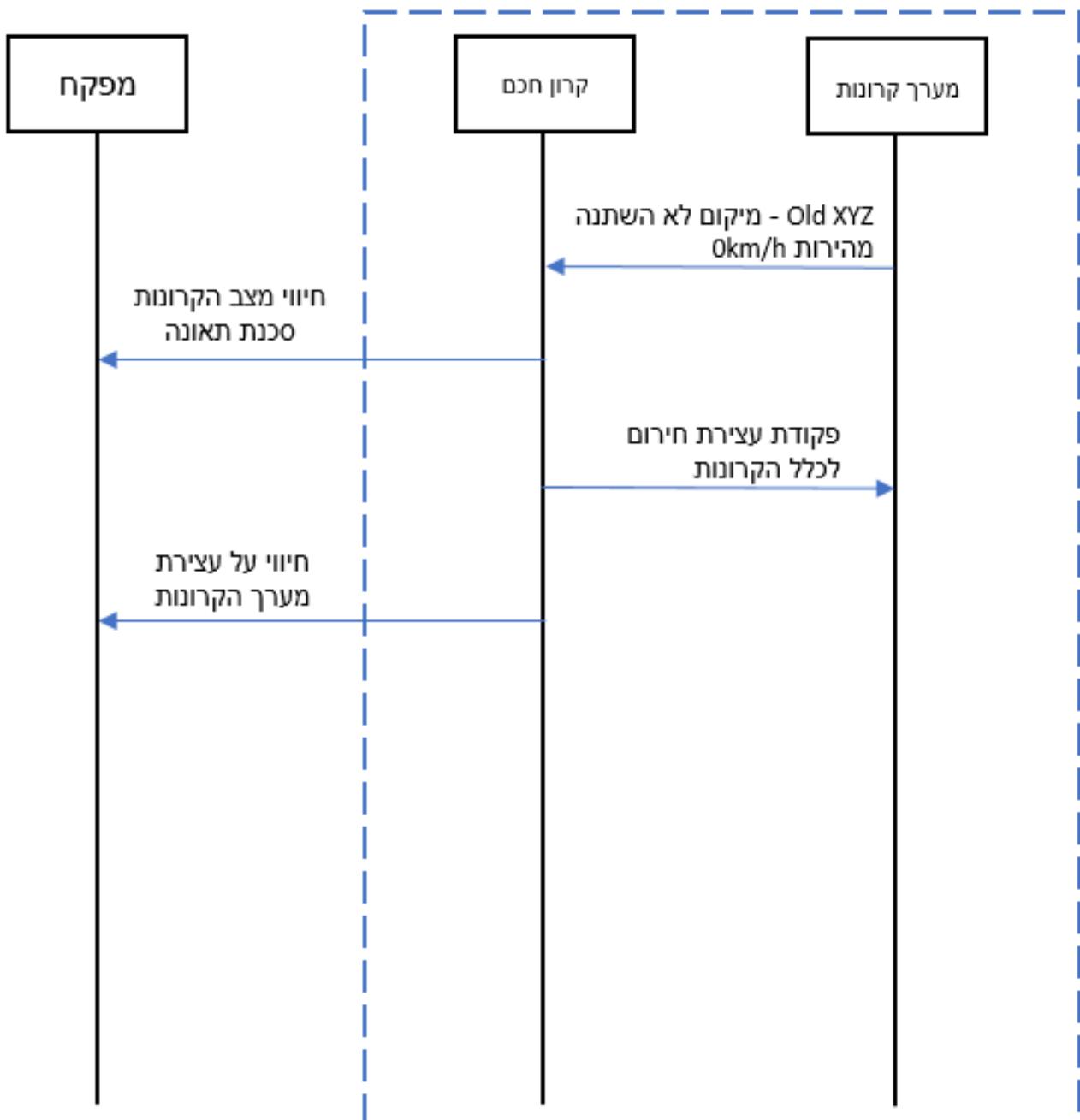
שינויי מיקום אחד הקرونוגות (סטיה/תקיעה) – מופע 1 – קופסה לבנה (שכבה 1):



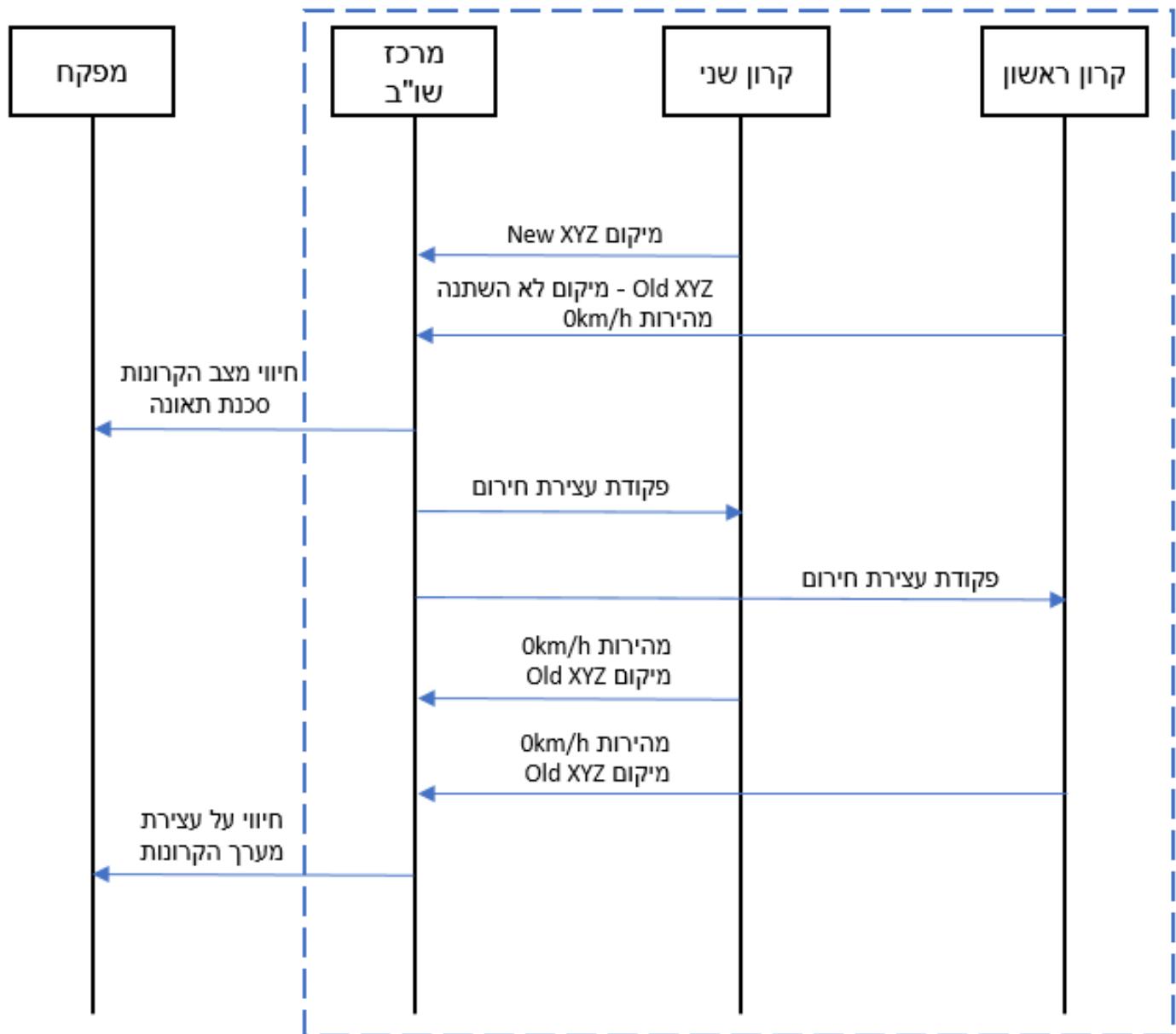
**שינויי מיקום אחד הקרןנות (סטיה/תקיעה) – מופע 1 – קופסה לבנה (שכבה 2):**



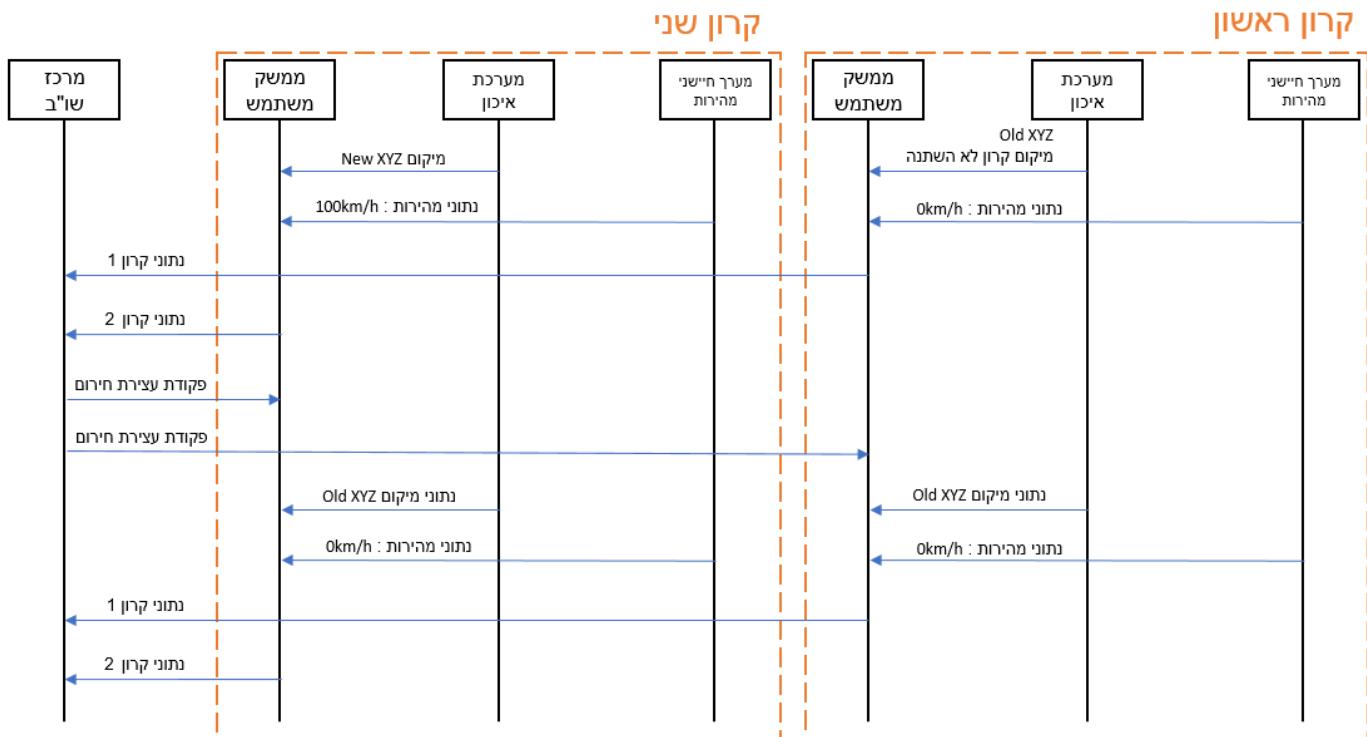
**שינויי מיקום אחד הקرونנות (סטייה/תקיעה) – מופע 2 – קופסה שחורה:**



**שינויי מיקום אחד הקرونות (סטיה/תקיעה) – מופע 2 – קופסה לבנה (שכבה 1):**



**שינויי מיקום אחד הקرونוגות (סטיה/תקיעה) – מופע 2 – קופסה לבנה (שכבה 2):**

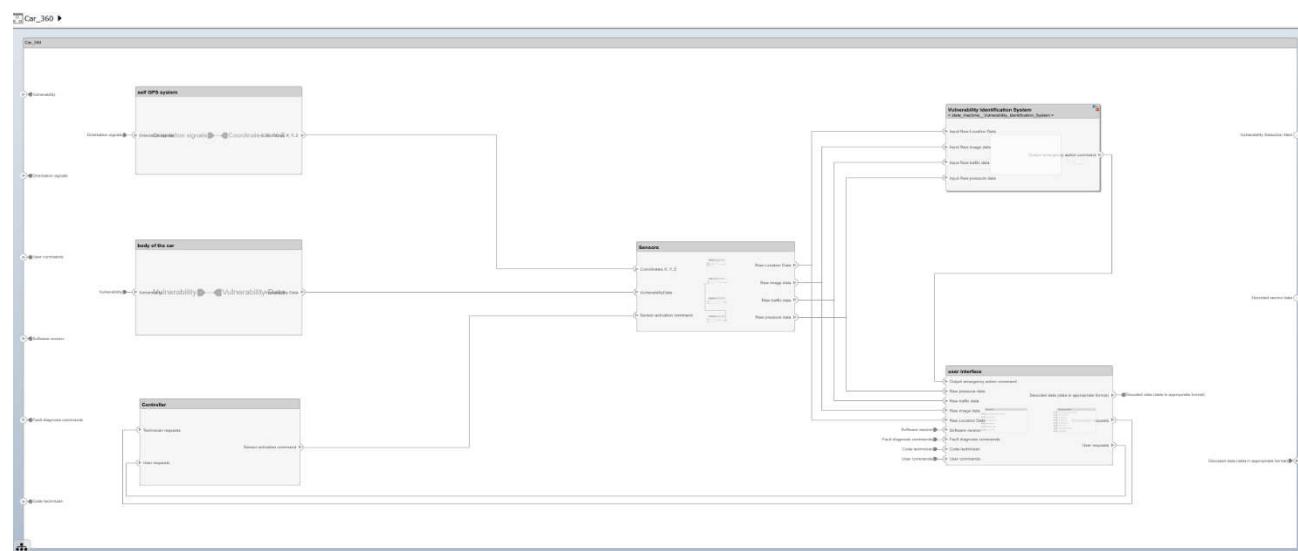


נספח ג: תיאור המידול והסימולציה - (מתוך סעיף 3.2)

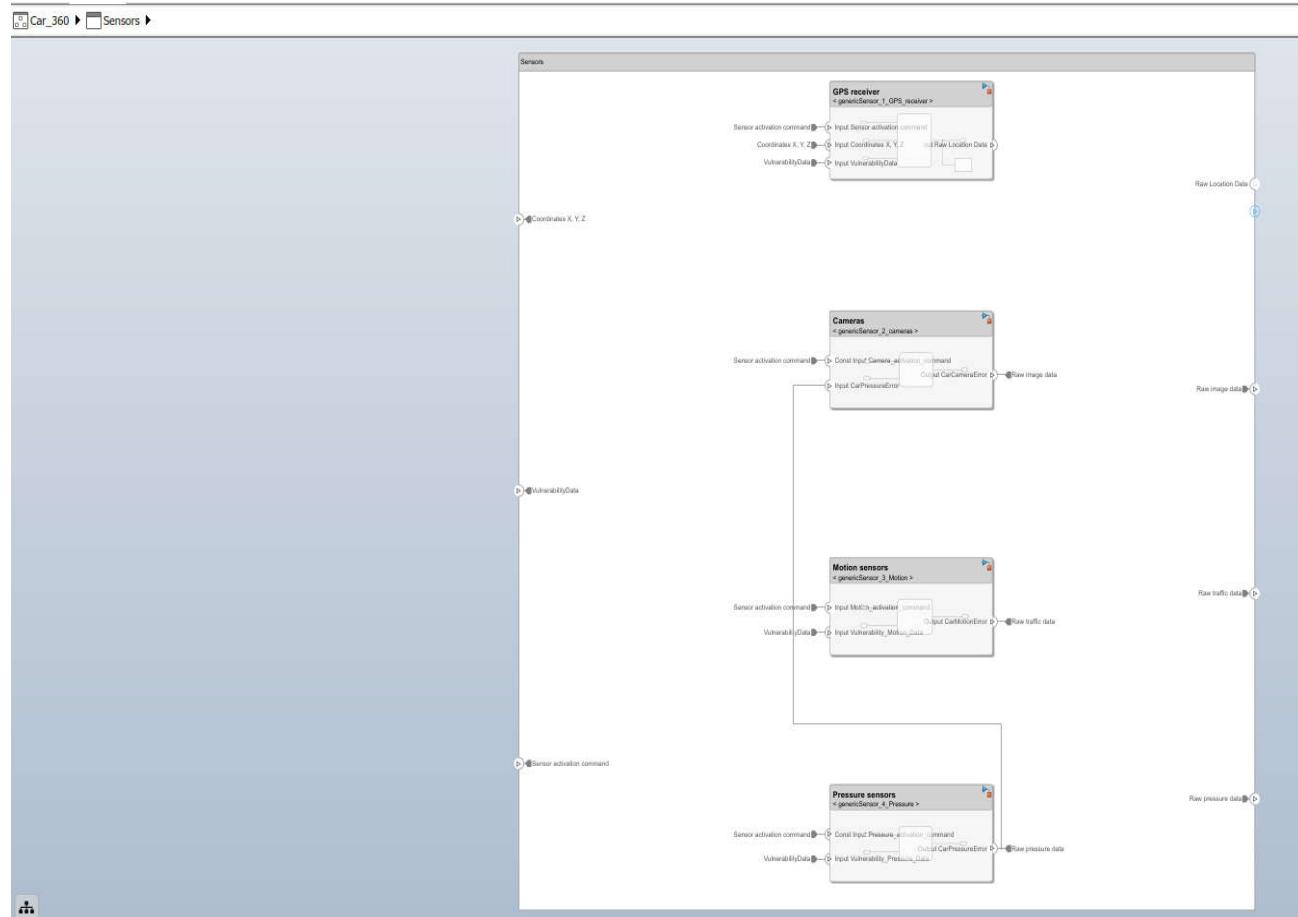
ג-1 - מידול וסימולציה באמצעות System Composer (3.2.5):

ניטור פגיעה ברכב 360:

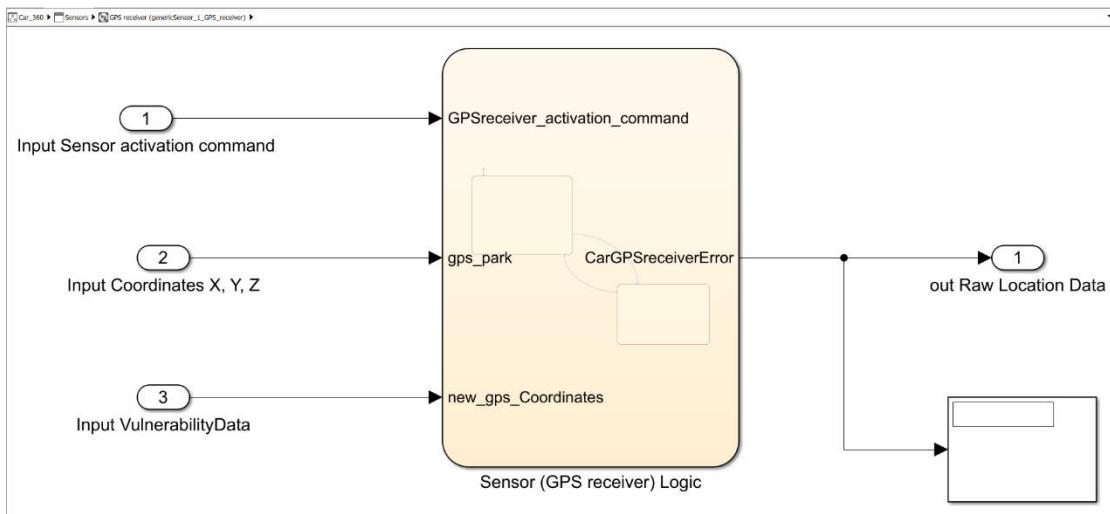
ארQUITקטורה כוללת של המערכת:



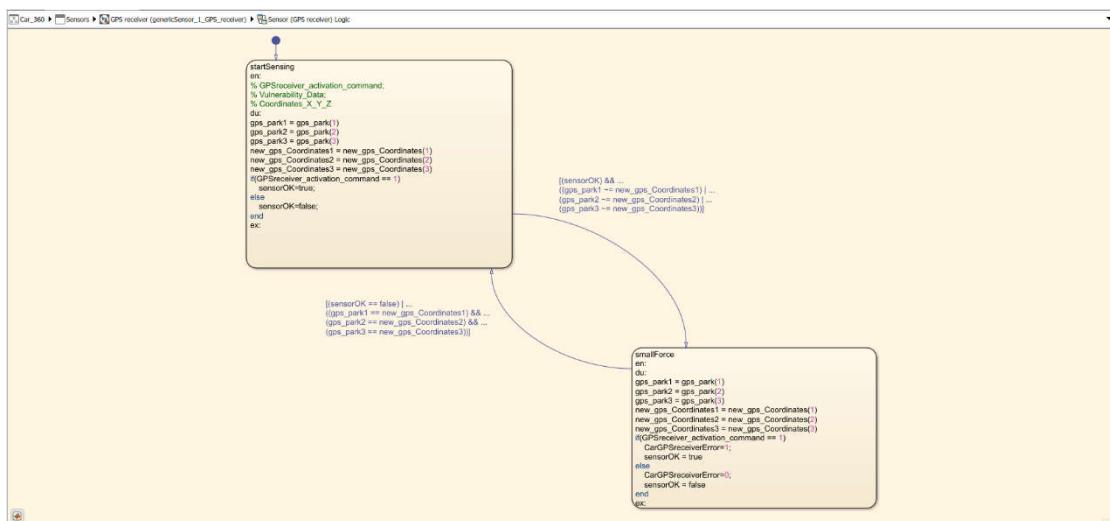
## ארQUITקטורת מערכת החלטות:



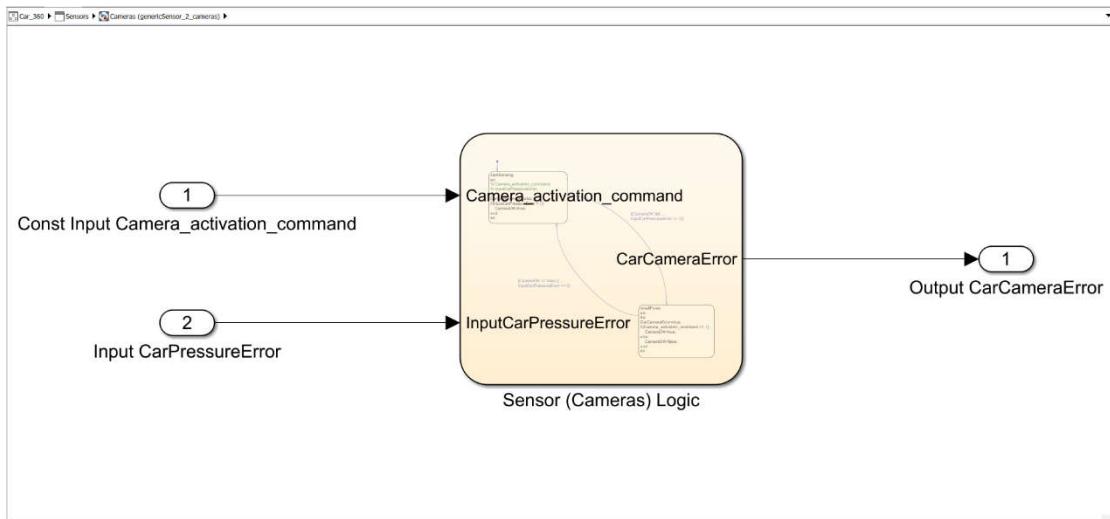
### ארכיטקטורת חיישן GPS:



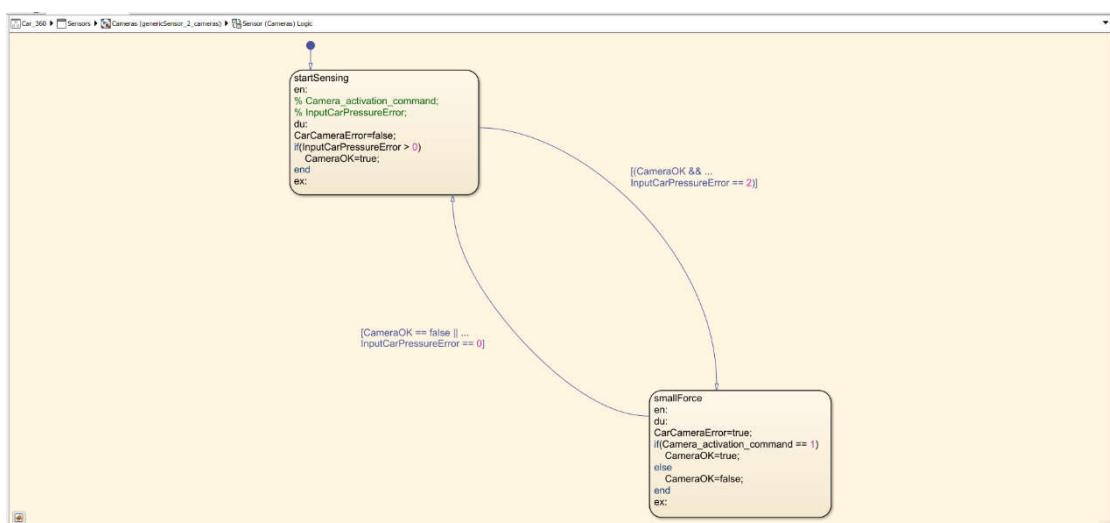
### GPS חיישן State flow



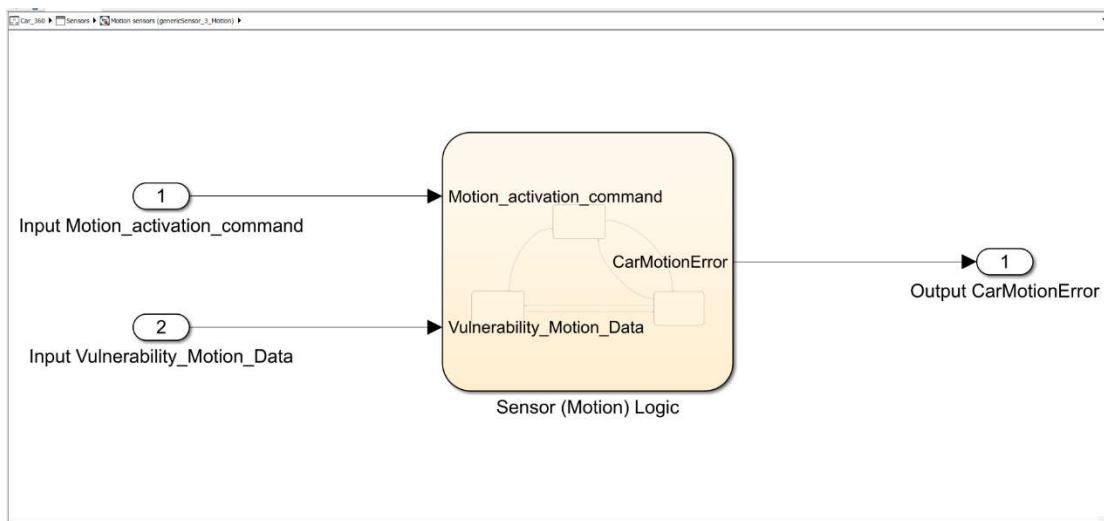
### ארQUITטורת חיישן המצלמה:



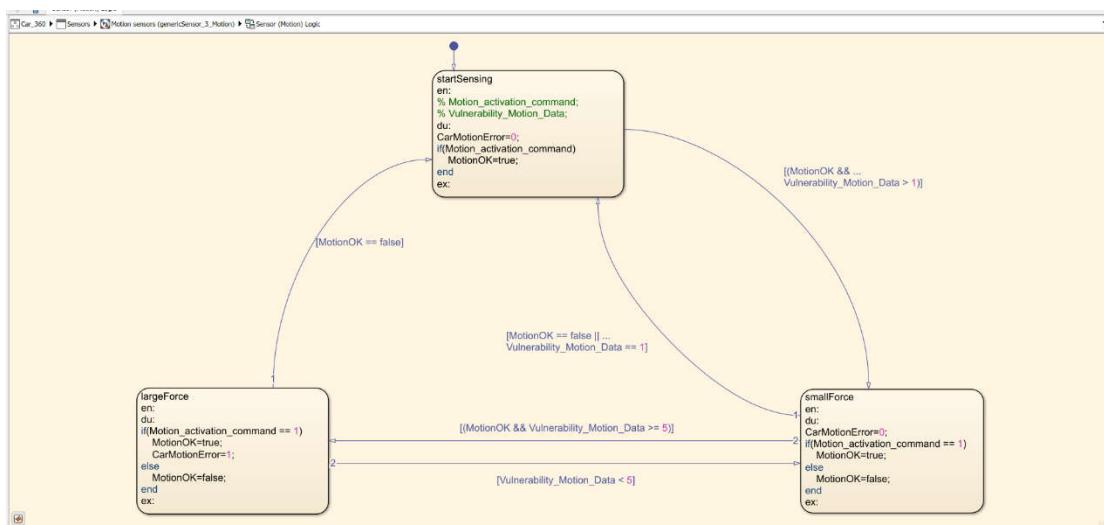
### חישן המצלמה: State flow



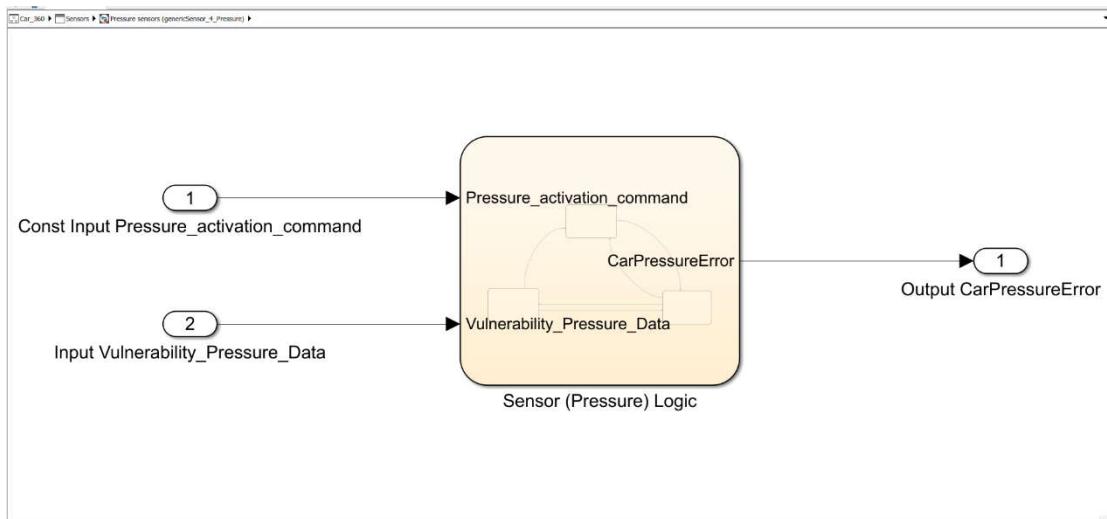
### ארQUITקטורת חיישן מהירות:



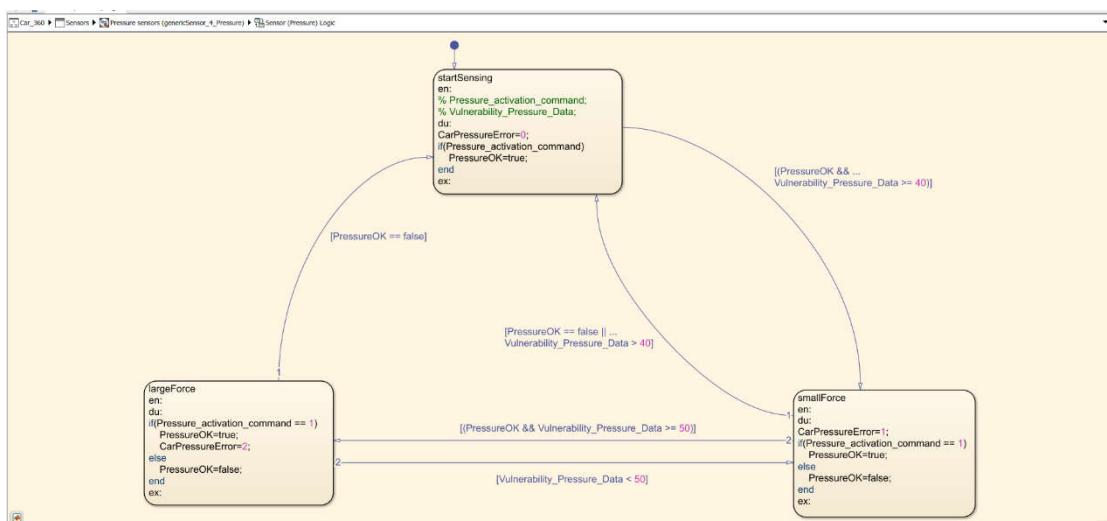
### חישון מהירות: State flow



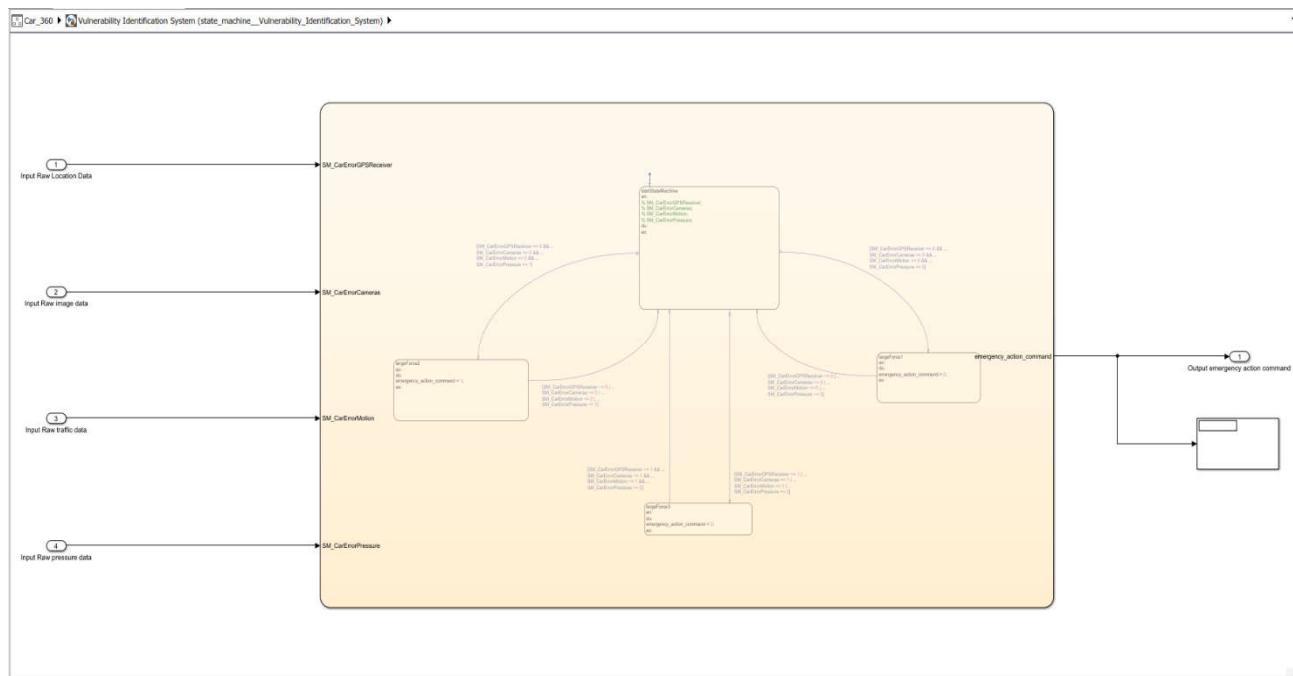
### ארכיטקטורת חיישן הלחץ:



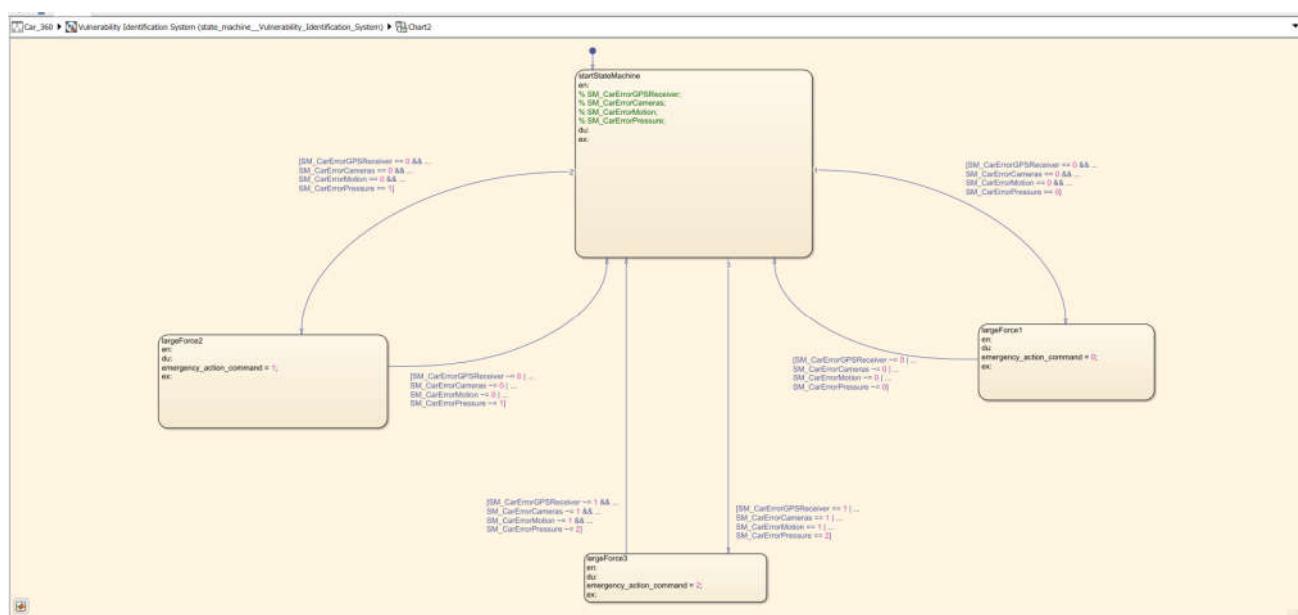
### חיישן לוחן State flow



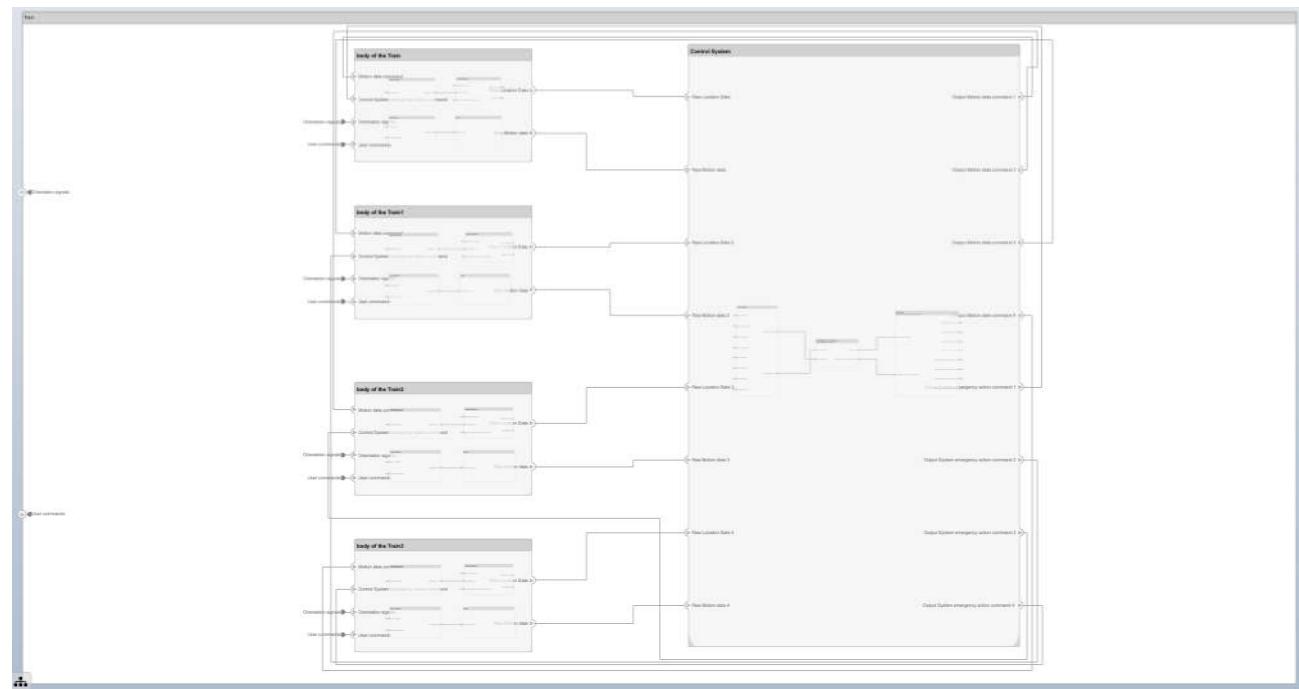
## ארQUITטורת מערך זהוי ופגיעות:



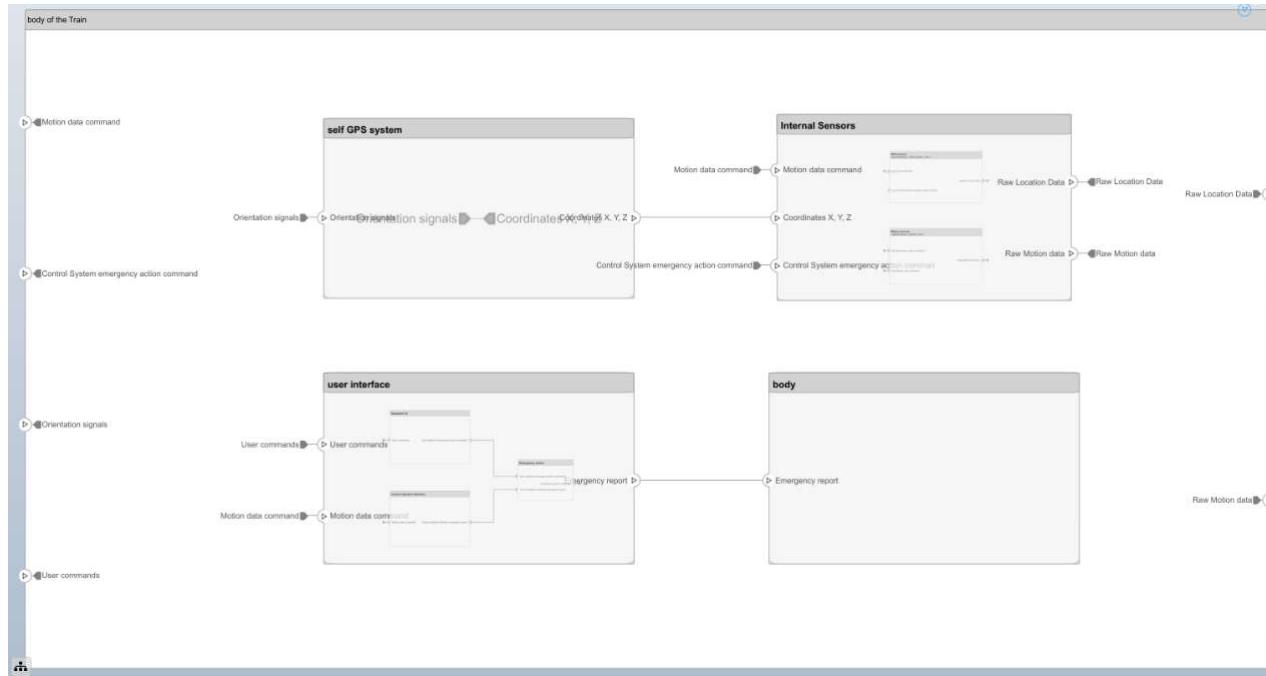
מערך זהוי ופגיעות: State flow



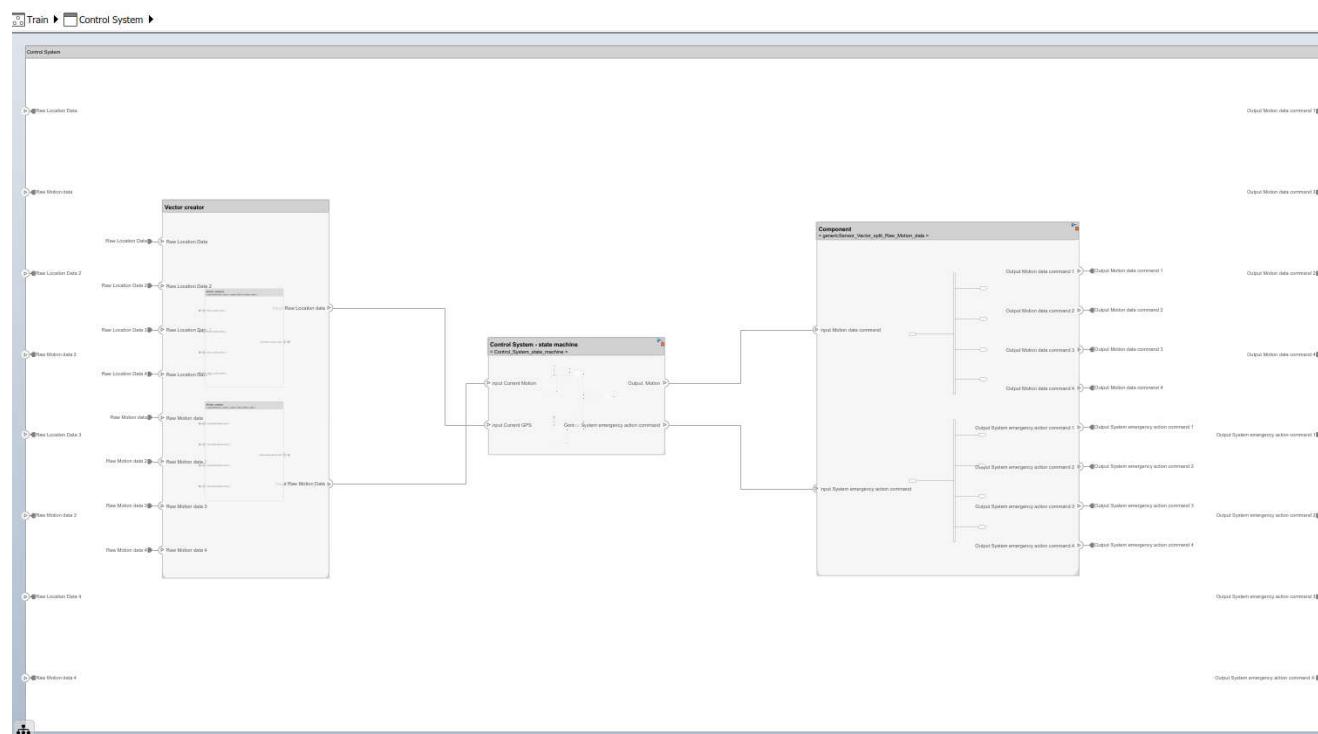
ארQUITקטורה כוללת של המערכת:



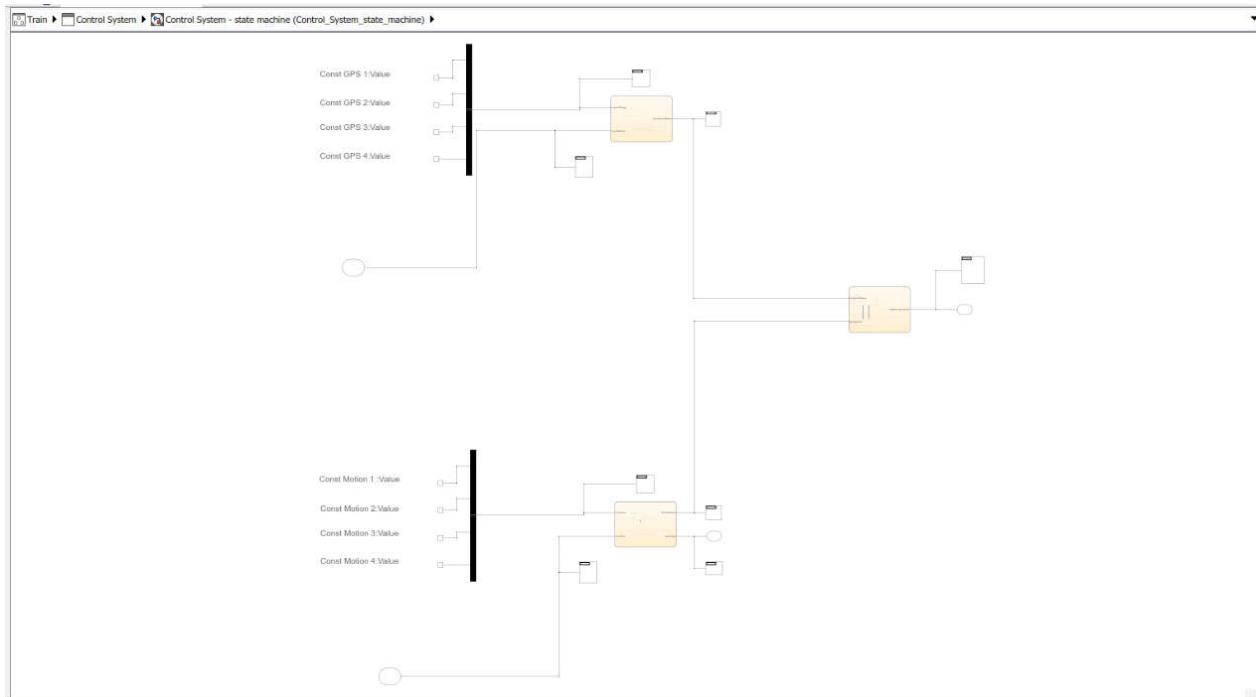
### ארQUITקטורת קרון אוטונומי בודד:



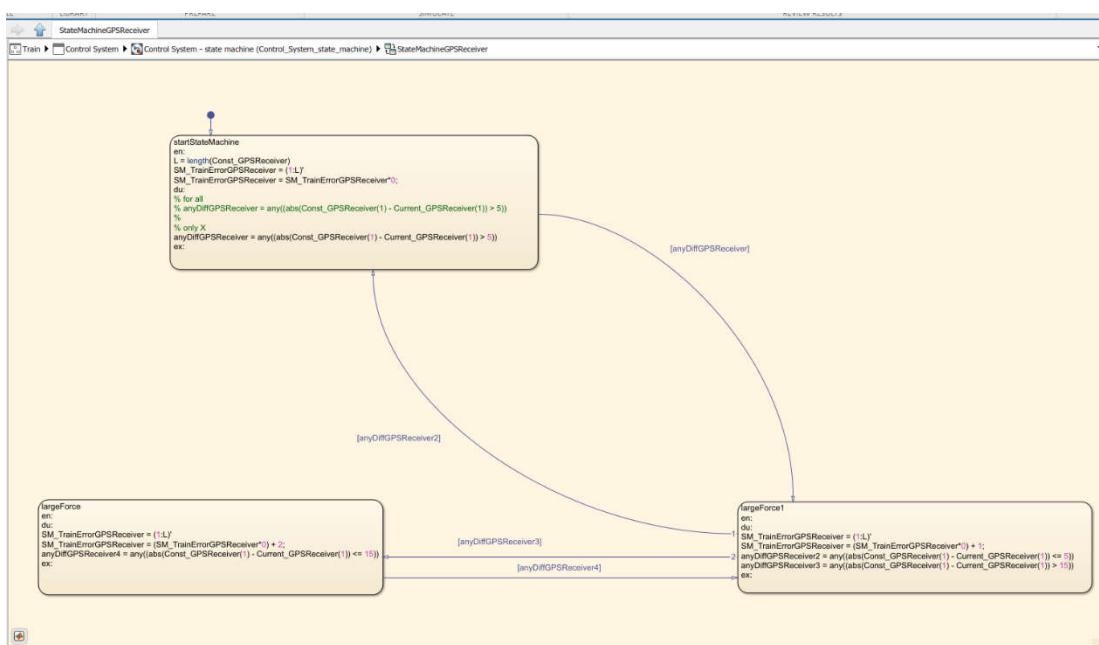
### ארQUITקטורת מערכת בקרה חכם ראשי:



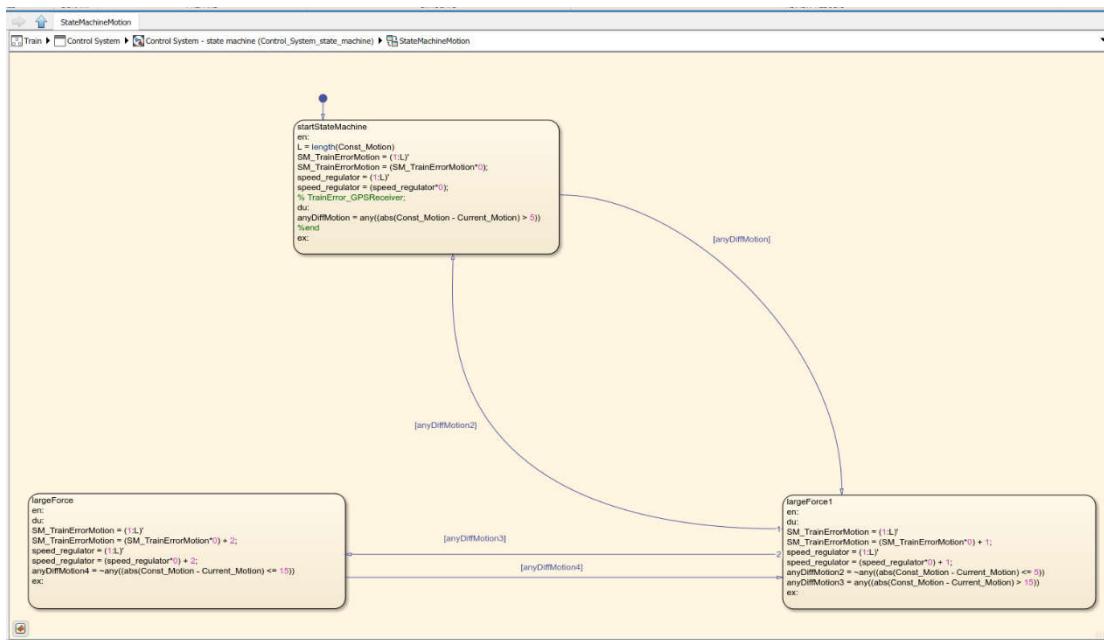
### ארQUITקטורת מערך בקרה חכם ראשית:



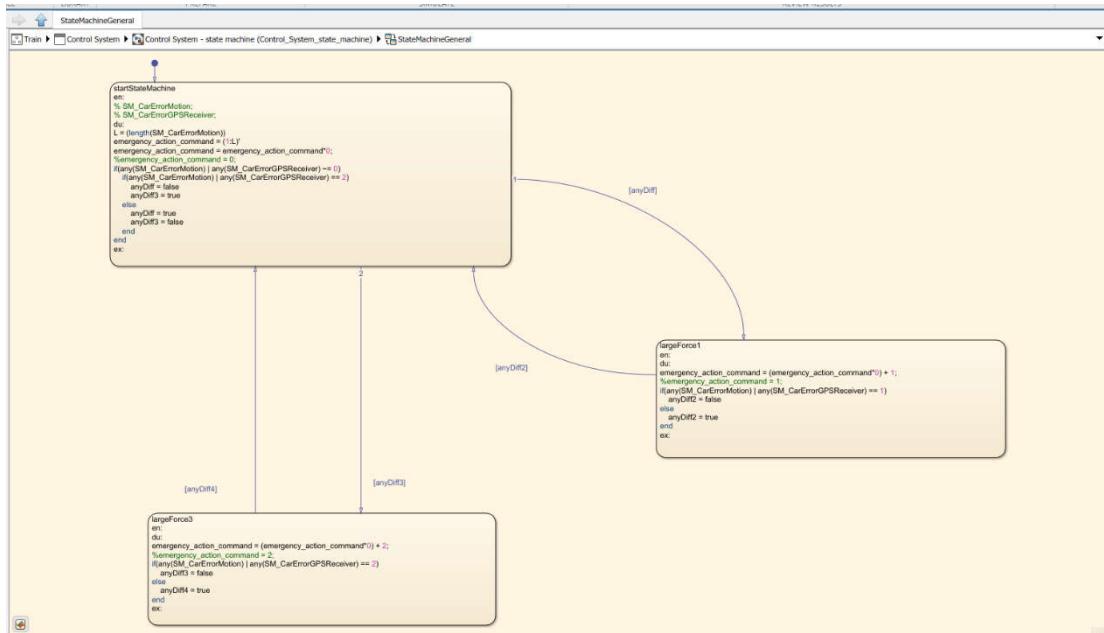
:GPS - מערך בקרה חכם State flow



## מערך בקרה חכם - מהירות State flow



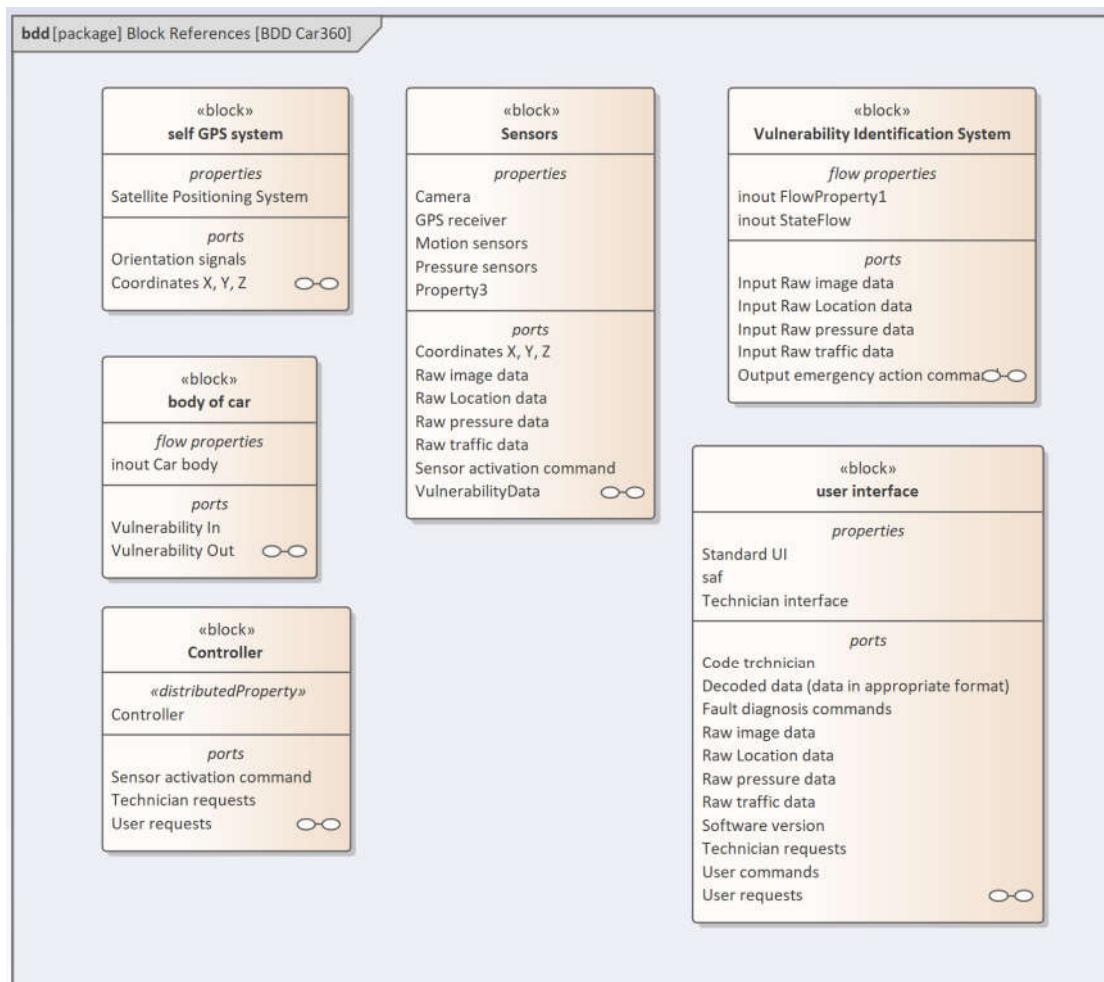
## מערך בקרה חכם - קבלת החלטות: State flow



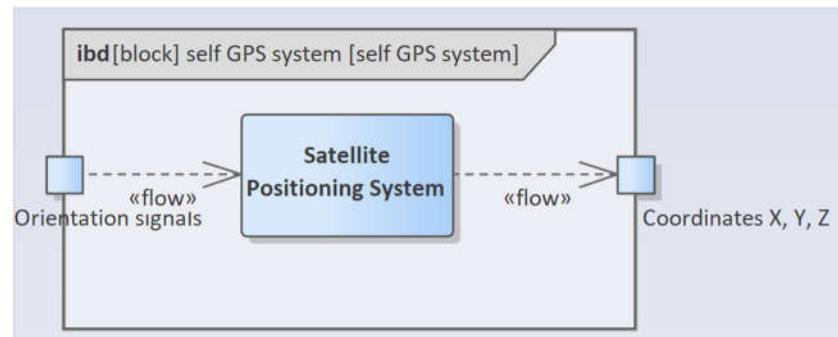
## ג-2 - מידול באמצעות Enterprise Architect (3.2.6)

ג-2.1 - ניתוח פגיעה ברכב 360

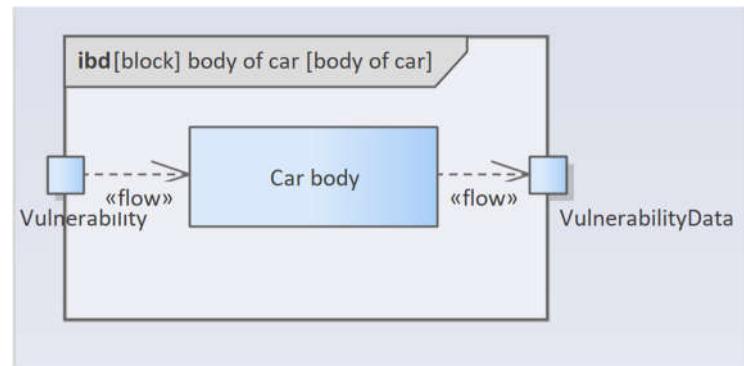
כלל המערכת של BDD



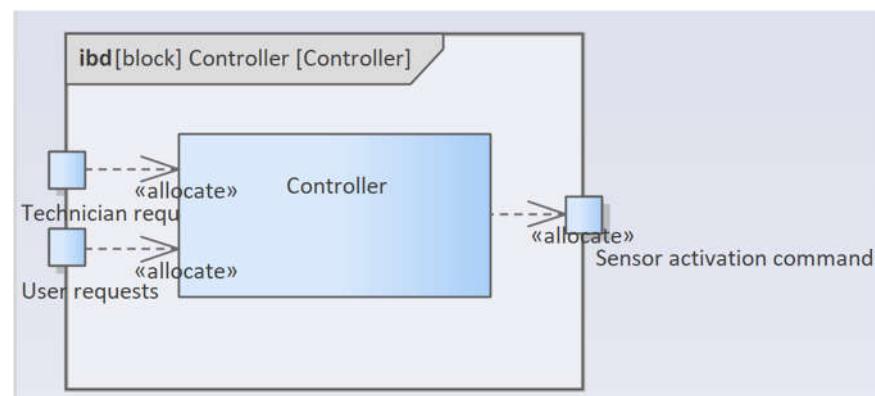
self GPS system של בлок IBD



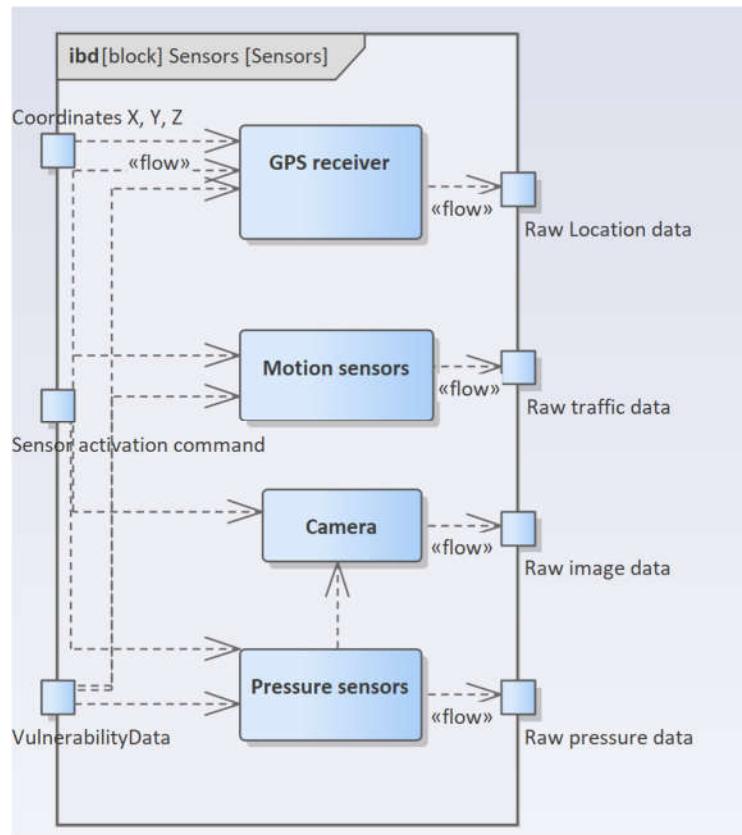
body of car של בלוק IBD



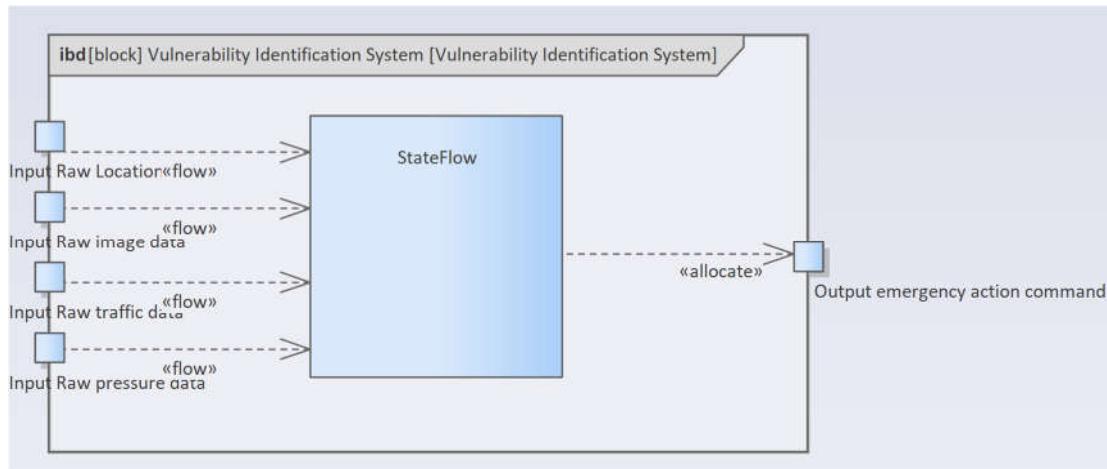
Controller של בלוק IBD



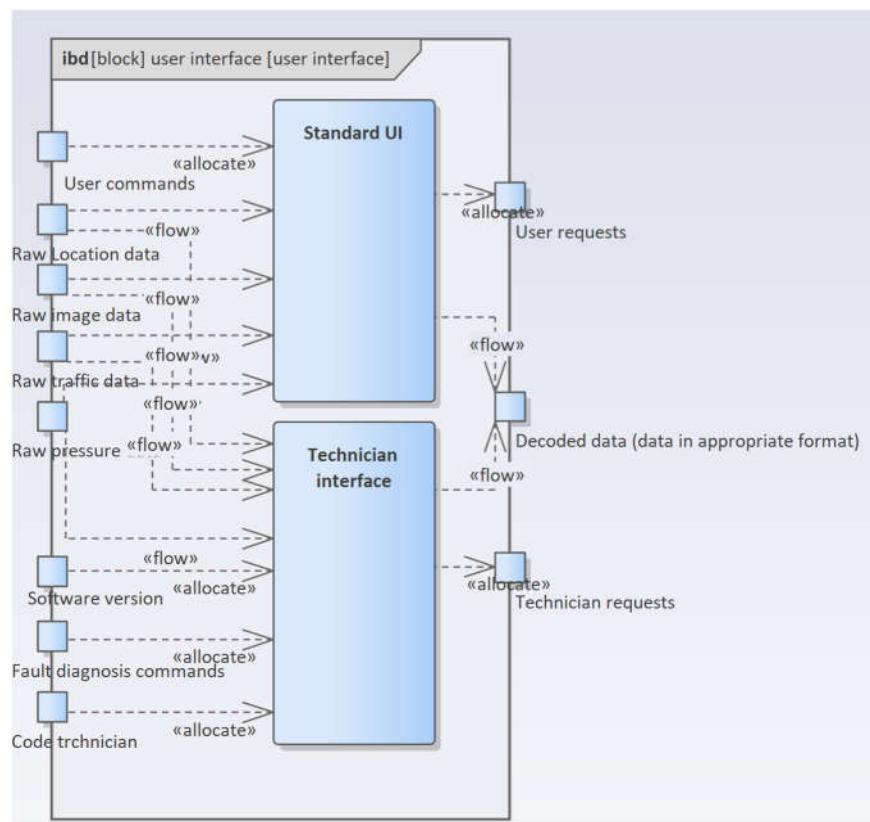
### Sensors של בлок IBD



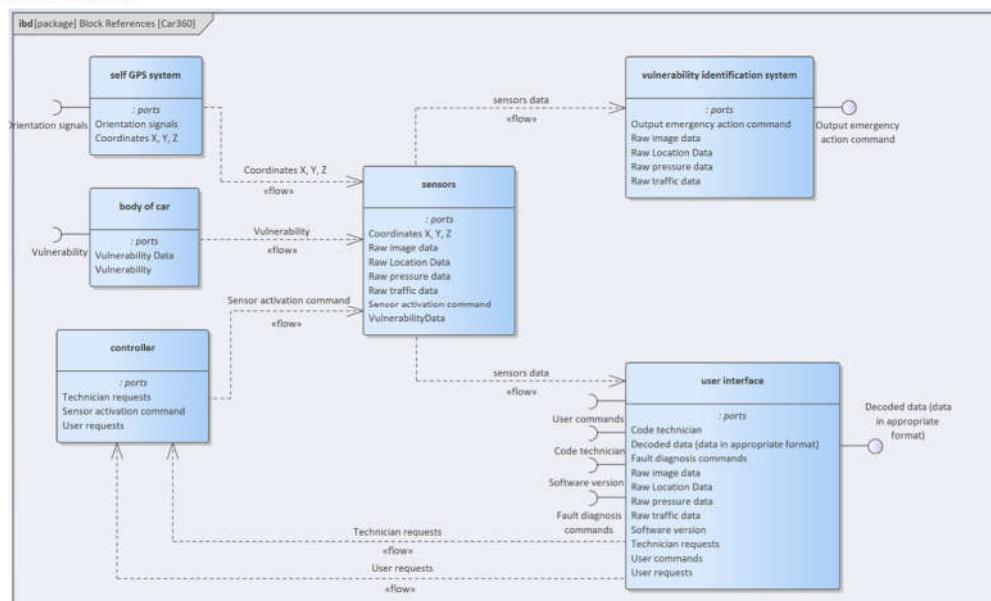
### Vulnerability Identification System של בлок IBD



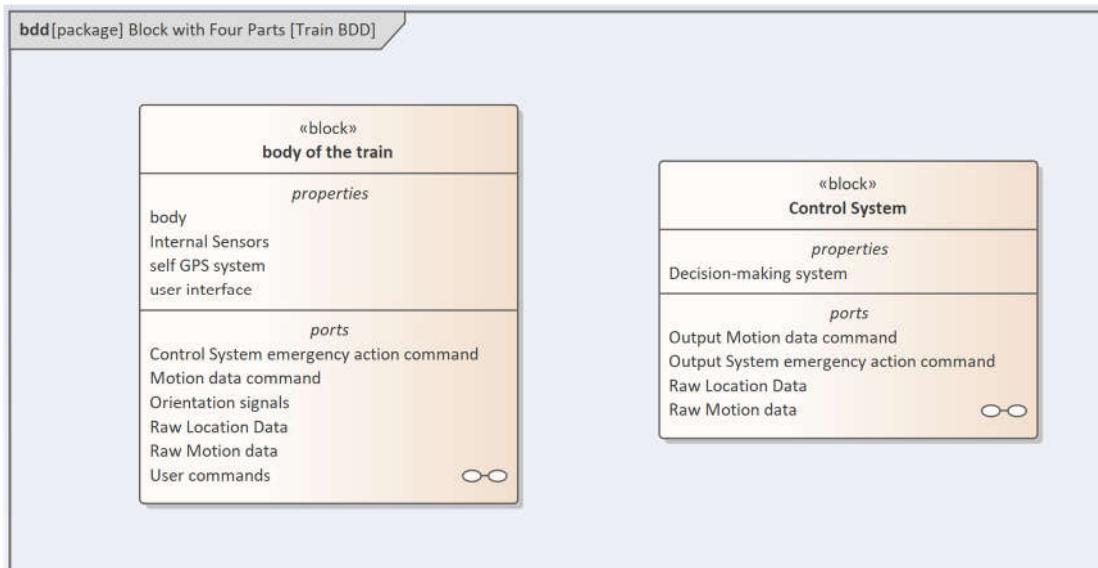
### user interface של בлок IBD



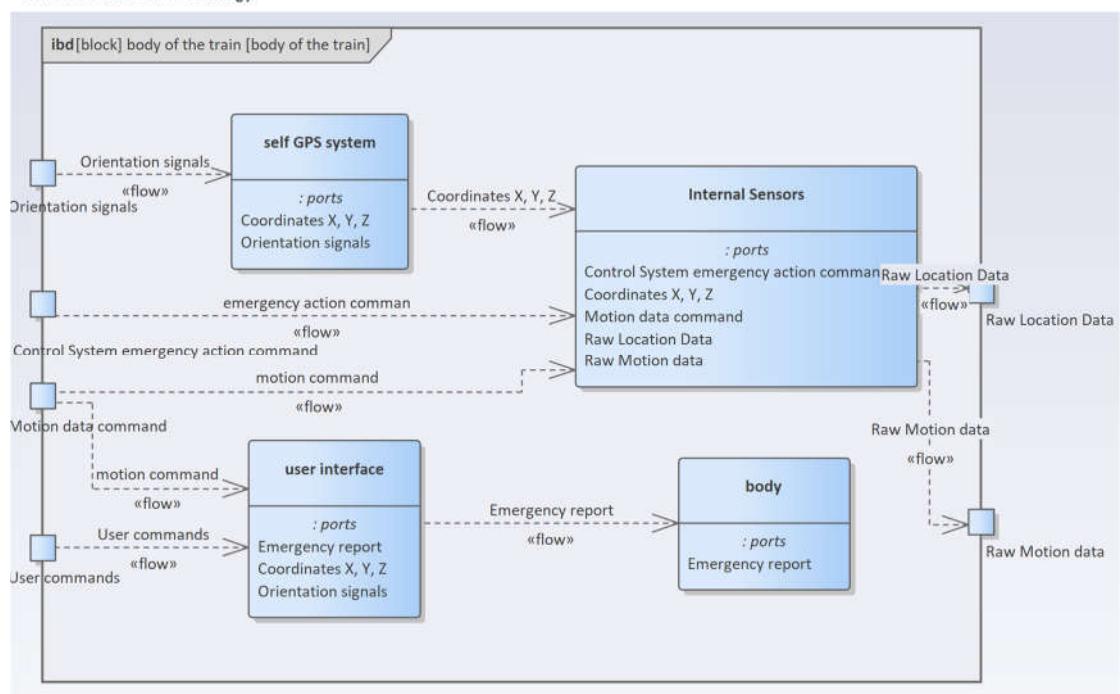
### כלל המערכת של IBD



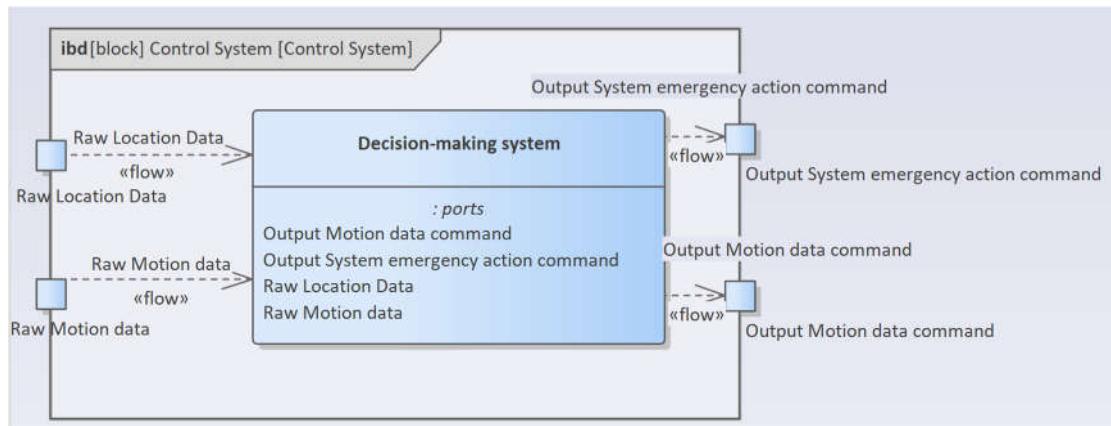
### ניהול מערכ קיוניות חכם: של כל המערכת BDD



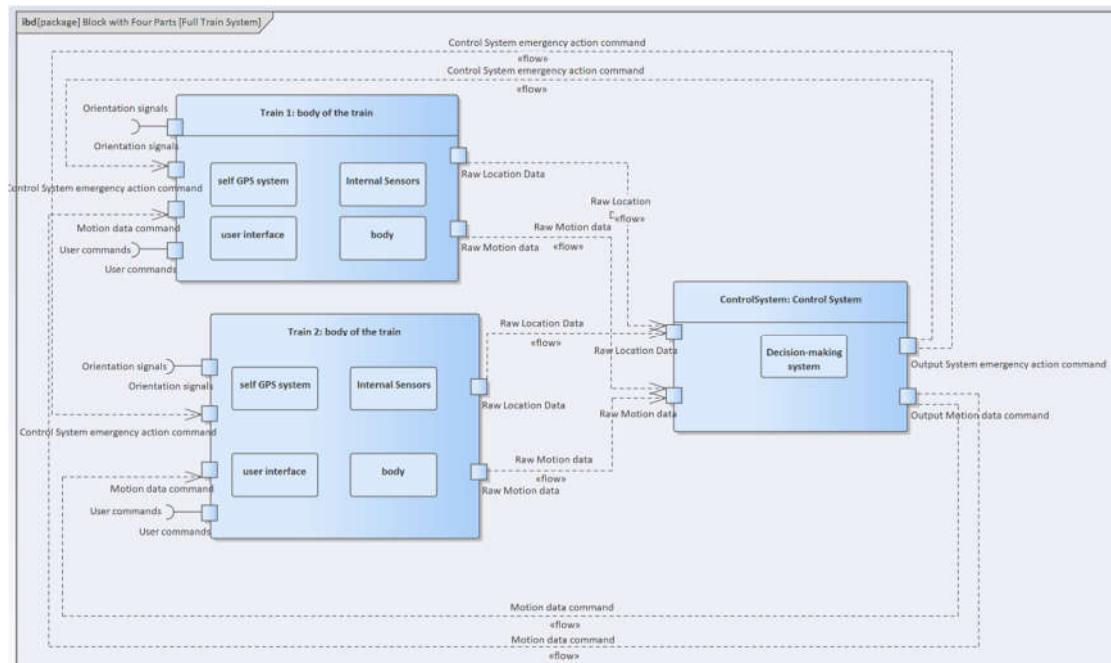
body of the train IBD של בלוק



### Control system של בлок IBD



### כלל המערכת של IBD



**נספח ד: אימות משקלים המדדים (מתיחס לסעיף 1.7)**

**ד-1 – שאלון המדדים**

שלום רב,

המחקר הנוכחי נערך במסגרת לימוד התואר השני של הסטודנטים אבי זגורி וליאור מנטיין בפקולטה לניהול טכנולוגיה במכון הטכנולוגי חולון.

שאלון זה עוסק במדדי המחקר "בחינת שימוש בשיטות ובכליים למידול מערכתי".

נודה לך אם תוכל להזכיר מזמן למלוי שאלון זה, הנתונים אשר יופקו משאלון זה ישמשו למטרת הפרויקט בלבד.

השאלון הוא אונימי.

**יש לענות על השאלון רק לאחר הצגת הפרזנטציה (פרקם: מבוא, סקר ספרות, מתודולוגיה והצגת הסימולציה) ע"י אבי זגורי ו/או ליאור מנטיין.**

1. האם אתה מסכימה/מרצונך החופשי לענות על שאלון זה?

- א. כן
- ב. לא

2. האם עני על שאלון זה בעבר?

- א. כן
- ב. לא

3. מהו גילך (בשנים)? \_\_\_\_\_

4. מהי עבודתך הנוכחי?

- א. שכיר/ה
- ב. עצמאי/ת
- ג. לא עובד/ת

5. מה תפקידך בחברה בה אתה עובד? \_\_\_\_\_

6. מהי רמת השכלתך?

- א. בעל תואר שלישי
- ב. בעל תואר שני
- ג. בעל תואר ראשון
- ד. תעודת בגרות
- ה. אחרת: \_\_\_\_\_

7. באיזה מקצוע רכשת את השכלתך? \_\_\_\_\_

8. לפניך סולם בן 6 דרגות. המספר 6 מציין את המدى החשוב ביותר, המספר 1 את המدى הצעיר ביותר.  
דרג את חשיבות המדרדים הבאים:

ديروج (1-6)	מדדים
	נוחיות ויזואלית
	ביצוע הגדרת חוזרת / עדכון הגדרה
	שינוי כניסה ויציאות מהמערכות השונות
	שינוי ארכיטקטורה באמצעות הזנת אלמנטים
	שפת מידול מוכרת לכל המשתתפים בפרויקט
	השפעה של שינויים באמצעות הפרויקט

9. לפניך טבלת מדדים, דרג את חשיבות המדדים באחוזים כבר שטר כל המדדים יהוו 100%.

משקלים	מדדים
	נוחיות ויזואלית
	ביצוע הגדרת חוזרת / עדכון הגדרה
	שינוי כניסה ויציאות מהמערכות השונות
	שינוי ארכיטקטורה באמצעות הזנת אלמנטים
	שפת מידול מוכרת לכל המשתתפים בפרויקט
	השפעה של שינויים באמצעות הפרויקט

10. לפניך רשימה של 6 ערכים. ציין לגבי כל אחד מהם באם הוא חשוב במידה רבה (4), חשוב במידה בינונית (3), חשוב במידה מועטה (2), כלל לא חשוב (1).

חשיבות (1-4)	מדדים
	נוחיות ויזואלית
	ביצוע הגדרת חוזרת / עדכון הגדרה
	שינויי כניסה ויציאה מהמערכות השונות
	שינוי ארכיטקטורה באמצעות הזנת אלמנטים
	שפה מידול מוכרת לכל המשתתפים בפרויקט
	השפעה של שינויים באמצעות הפרויקט

**ד-2 – תוצאות שאלון המדרדים**

מדד'ים	ויזואלית/ נוחיות	ביצוע הגדרת/ עדכן הגדרה	שינוי כנישות ויציאות מהמערכות השונות	שינויי ארכיטקטורה באמצעות הצעת אלמנטים	שפת מידול מוכרת לכל המשתתפים בפרויקט	שפת מידול מוכרת באמצע הפרויקט	השפעה של שינויים באמצע הפרויקט
סקר 1	15	15	15	15	20	5	30
סקר 2	15	25	5	5	5	10	40
סקר 3	25	10	0	15	10	10	40
סקר 4	20	20	10	10	10	10	30
סקר 5	25	25	5	15	15	15	20
סקר 6	30	20	0	10	10	15	25
סקר 7	20	20	10	10	10	10	30
סקר 8	20	20	10	10	10	10	30
סקר 9	20	20	10	10	10	10	30
סקר 10	20	20	10	10	10	10	30
ממוצע	21	19.5	7.5	11.5	10	10	30.5
משקלים בפועל	20	20	10	10	10	10	30