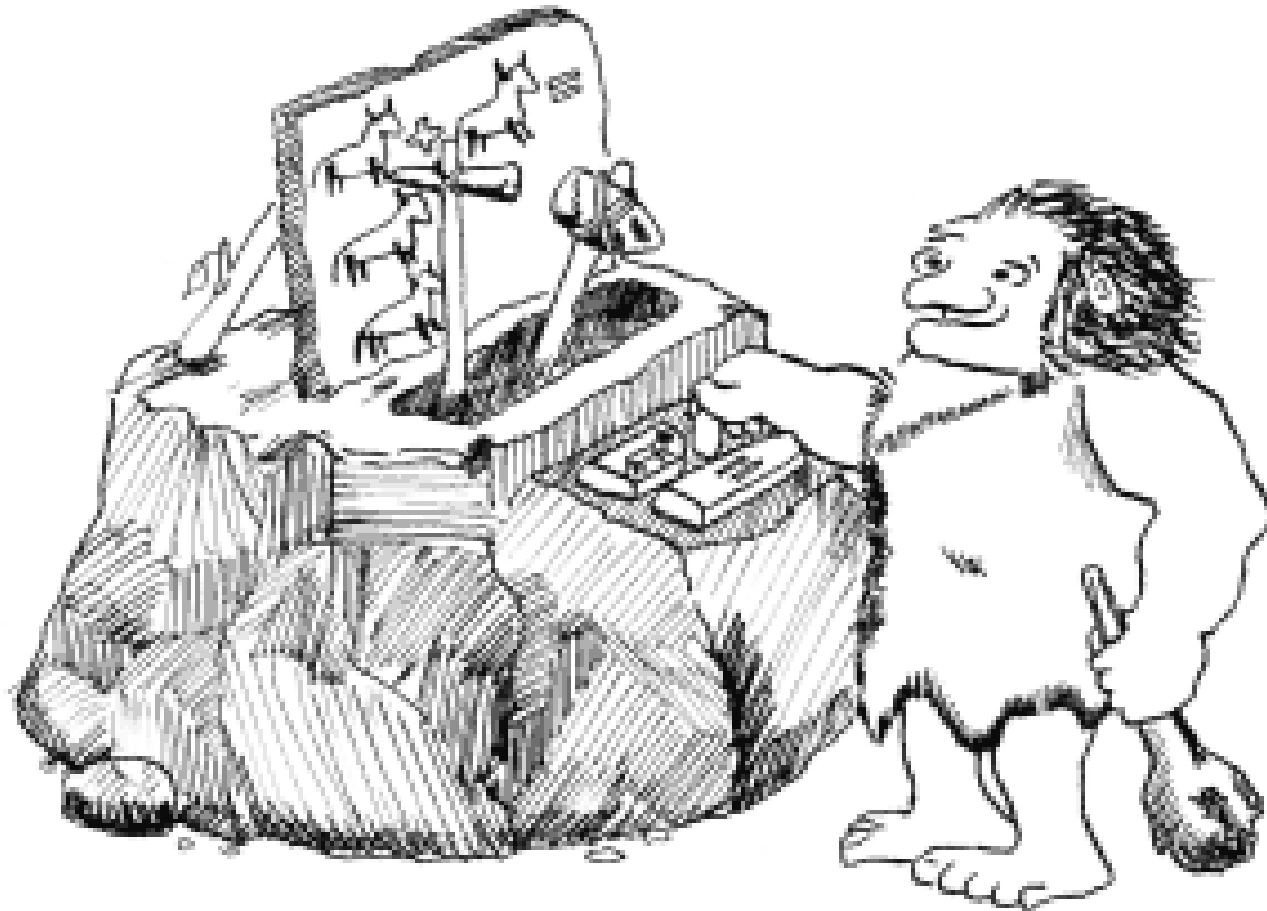


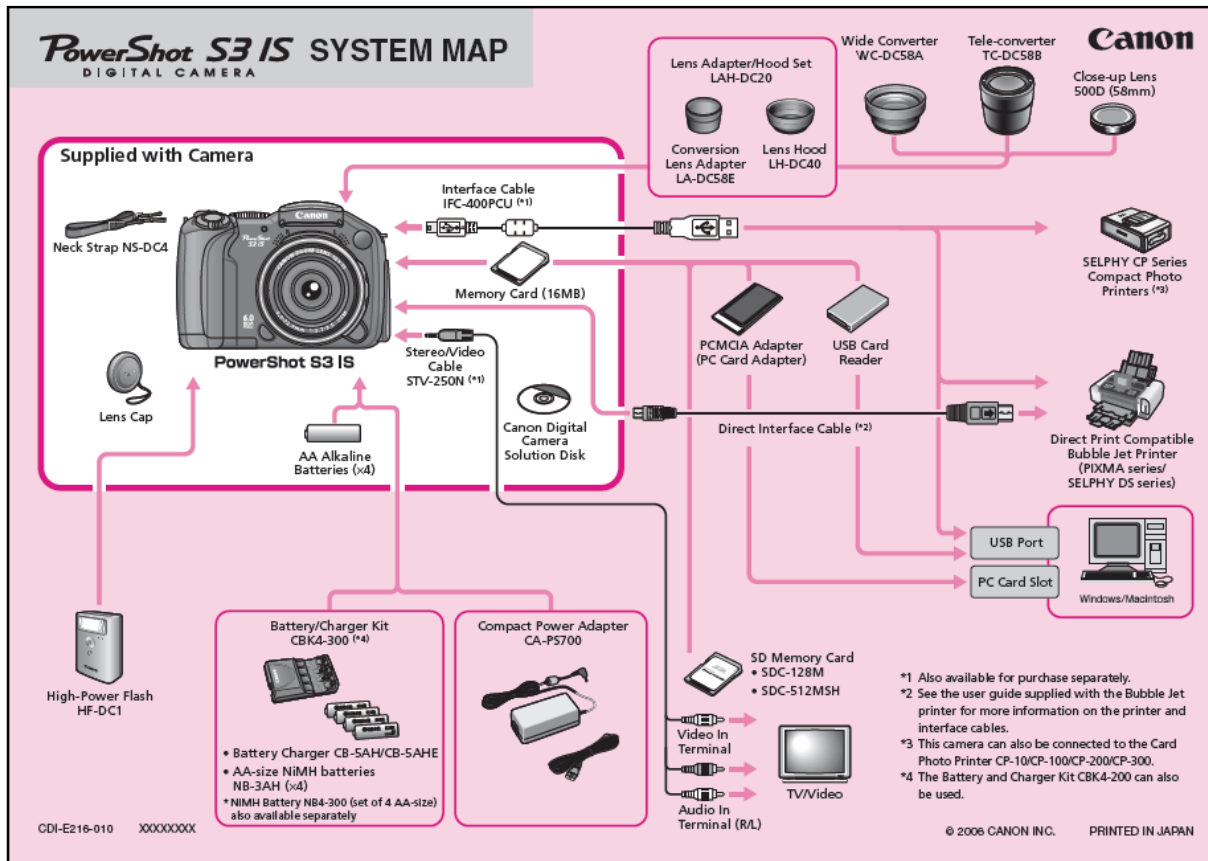
ארכיטקטורת מערכת ותוכנה

System and Software Architecture



ארכיטקטורה ותצורות

- תצורה (configuration) – מבנה המערכת (רכיבים, כמויות, קשרים) בהקשר ספציפי (שימוש, שלב במחזור חיים וכו')
- ארכיטקטורה – המבנה העקרוני של המערכת, ממנו נגזרות תצורות השונות



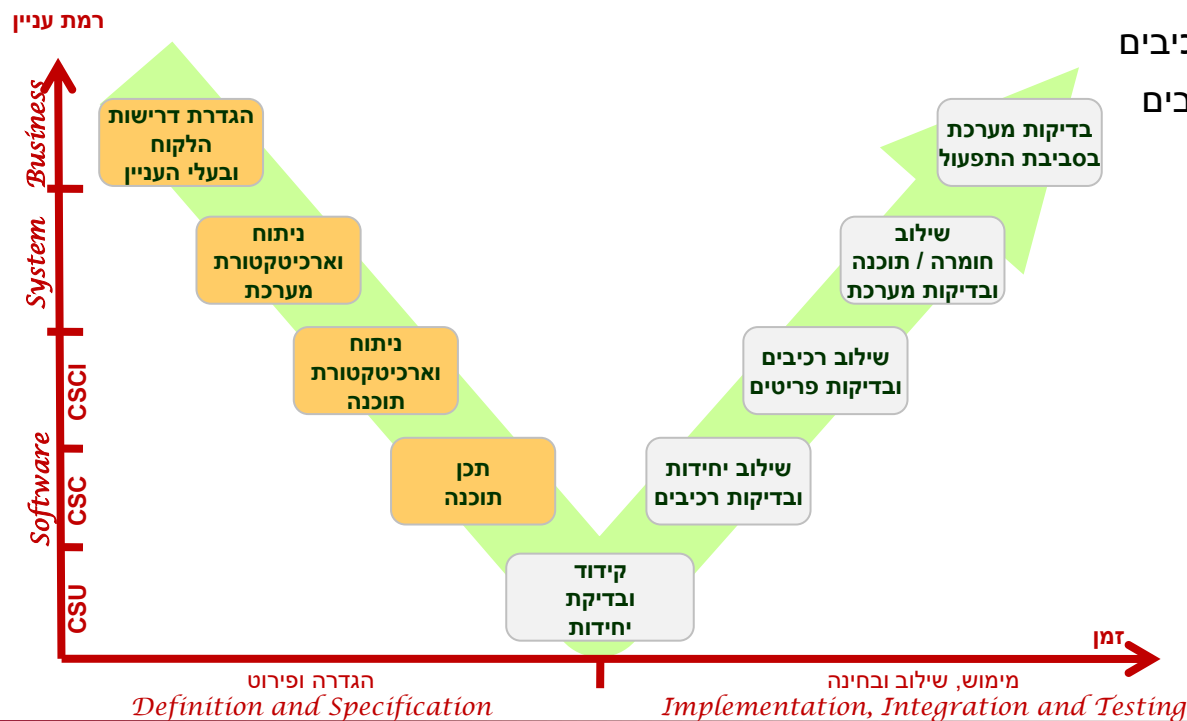
פעילות הגדרת הארכיטקטורה

- ניתן להגדיר ארכיטקטורה בכל רמת עניין

- רמת הארגון (מבנה ארגוני ופריסת מערכות)
- רמת מערכת המיחשוב ("קופסאות" החומרה ויישומי התוכנה המותקנים בהם)
- רמת פריט/יישום התוכנה (הרכיבים ממנו הוא בנוי)
- רמת רכיב התוכנה (מודולי הקוד המרכיבים אותו)

- פעילות הגדרת ארכיטקטורה

- זיהוי מרכיבים, בהתאם לרמה
- הקצאת פונקציונאליות למרכיבים
- הגדרת ממשקים בין המרכיבים



ארכיטקטורה של מערכת עתירת תוכנה



• ארכיטקטורה פונקציונאלית / לוגית

- רכיבים פונקציונאליים / לוגיים
 - רכיבי שימוש (בזמן ריצה)
 - חלקי תוכנה אופרטיביים המבצעים משימות בזמן שהמערכת פעילה
 - » לדוגמה: יישומים, דרייברים, רכיבי תקשורת, DLL, GUI, ...
 - רכיבי מימוש (בזמן כתיבה)
 - חלקי קוד מקור (source code) מהם בונים את רכיבי השימוש
 - » לדוגמה: מודולים בשפת תכנות, ספריות, open source, ...
- ממשקים פונקציונאליים / לוגיים
 - המידע המועבר בין הרכיבים הפונקציונאליים
 - לדוגמה: אותות בקרה, פקודות, נתונים, הודעות, ...

• ארכיטקטורה פיזית

- רכיבים פיזיים
 - "קופסאות" חומרה
 - לדוגמה: מעבדים, התקני אחסון, התקני תקשורת, ...
- ממשקים פיזיים
 - חיבורים בין רכיבים פיזיים המאפשרים להעביר באמצעותם מידע
 - כבלים, סיבים אופטיים, שידור/קליטה אלקטרומגנטית, ...

המקרה האופייני
למערכות משובצות
רב-תחומיות

תיכון ארכיטקטורה מערכתית - ממה מתחילים?

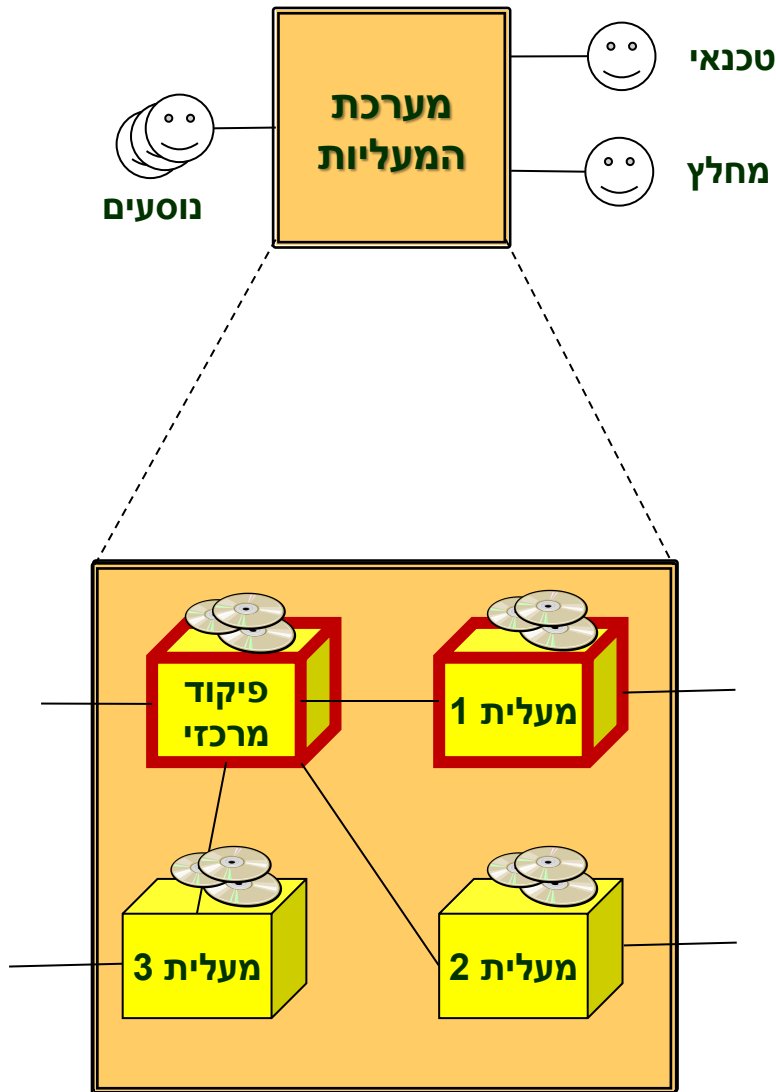
- מקרה א': הארכיטקטורה הפיזית קיימת / נקבעה מראש

- מזהים רכיבים וממשקים לוגיים
- שוקלים חלופות של הקצאת רכיבים לוגיים לרכיבים הפיזיים בארכיטקטורה הנתונה
- בונים ארכיטקטורה לוגית

- מקרה ב': הארכיטקטורה הפיזית טרם נקבעה

- בונים ארכיטקטורה לוגית
- שוקלים חלופות של בחירת רכיבים וממשקים פיזיים
- בונים ארכיטקטורה פיזית

המקרה האופייני
למערכות מידע



- עד עתה הגדרנו את המערכת כ"קופסה שחורה"

- השלב הבא – פירוק למרכיבים והגדרת מבנה והתנהגות

- מבנה (ארכיטקטורה פיזית) – על פי האפיון הטכני

- פריטי חומרה (מחשבים, התקנים)

- HWCI = HardWare Configuration Item

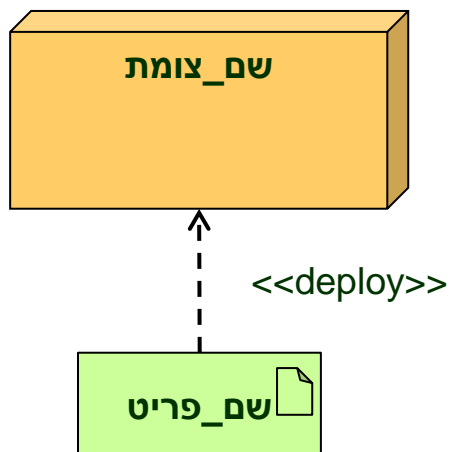
- פריטי תוכנה / יישומים

- CSCI = Computer Software Config. Item

- מיקום התוכנה על גבי החומרה ("התקנה")

- תקשורת פיזית / פרוטוקולים

ארכיטקטורת מערכת מחשוב ב-UML



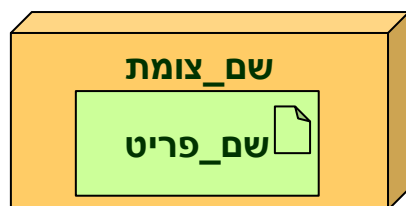
- צומת (Node)

– אובייקט פיזי פעיל המייצג משאב חישובי, אשר בדרך כלל כולל זיכרון ולרוב גם יכולת עיבוד.

- פריט (Artifact)

– מרכיב מידע פיזי שנוצר בתהליך הפיתוח או נמצא בשימוש המערכת בזמן הפעלתה

סימול אלטרנטיבי



- פריסה (Deployment)

– הקצאה (התקנה) של פריטי התוכנה בצמתי החומרה

חומרה – צמתים אופייניים

- מחשבים

- שרתים
- מחשבי קצה
- מיקרו-פרוססורים

- התקני איחסון

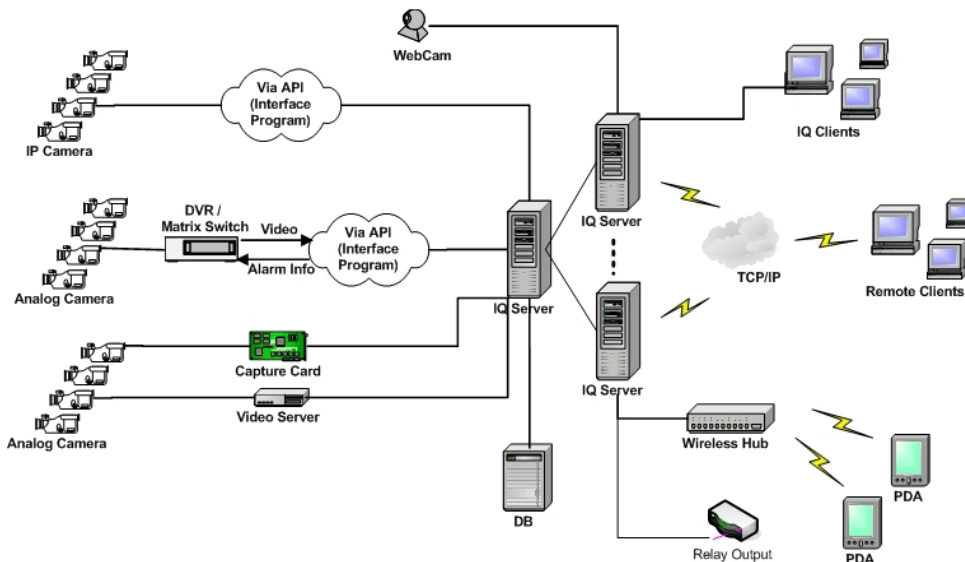
- כוננים
- התקני זכרון חיצוני

- התקני תקשורת פיזית

- מודמים
- נתבים

- ציוד היקפי (Peripherals)

- מדפסות, סורקים, ...
- התקני תצוגה
- חיישנים



תוכנה – פריטים אופייניים

- **תוכנת תשתית**

- מערכות הפעלה ותקשורת
- יישומים סטנדרטיים (למשל אופיס)

- **קבצי ריצה**

- .jar , .dll , .exe
- דרייברים

- **הגדרות**

- קבצי התקנה / registry
- תבניות

- **פריטי נתונים**

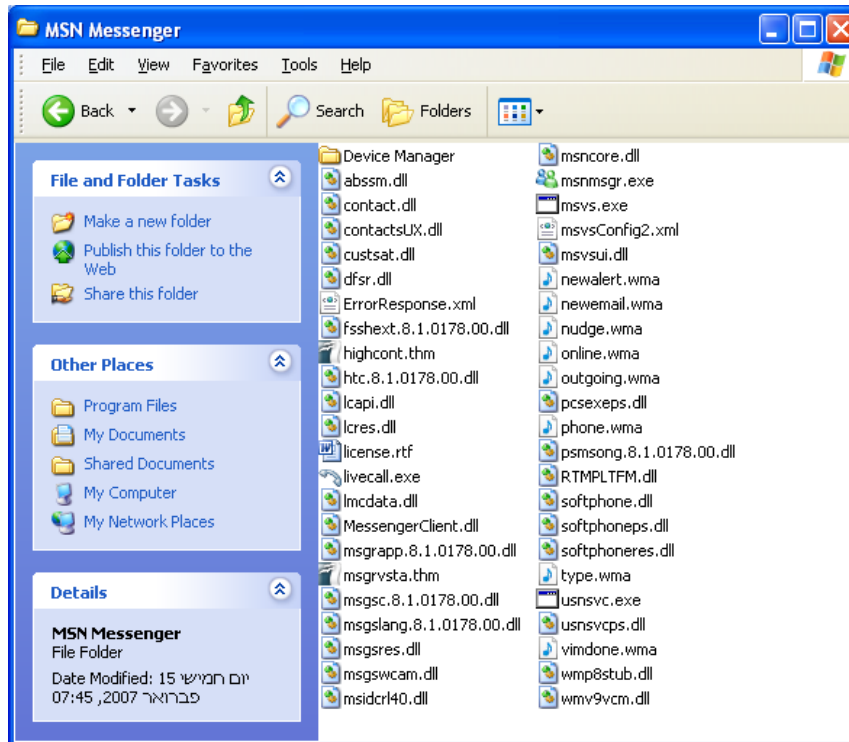
- קבצי נתונים
- בסיסי נתונים

- **פריטי מדיה**

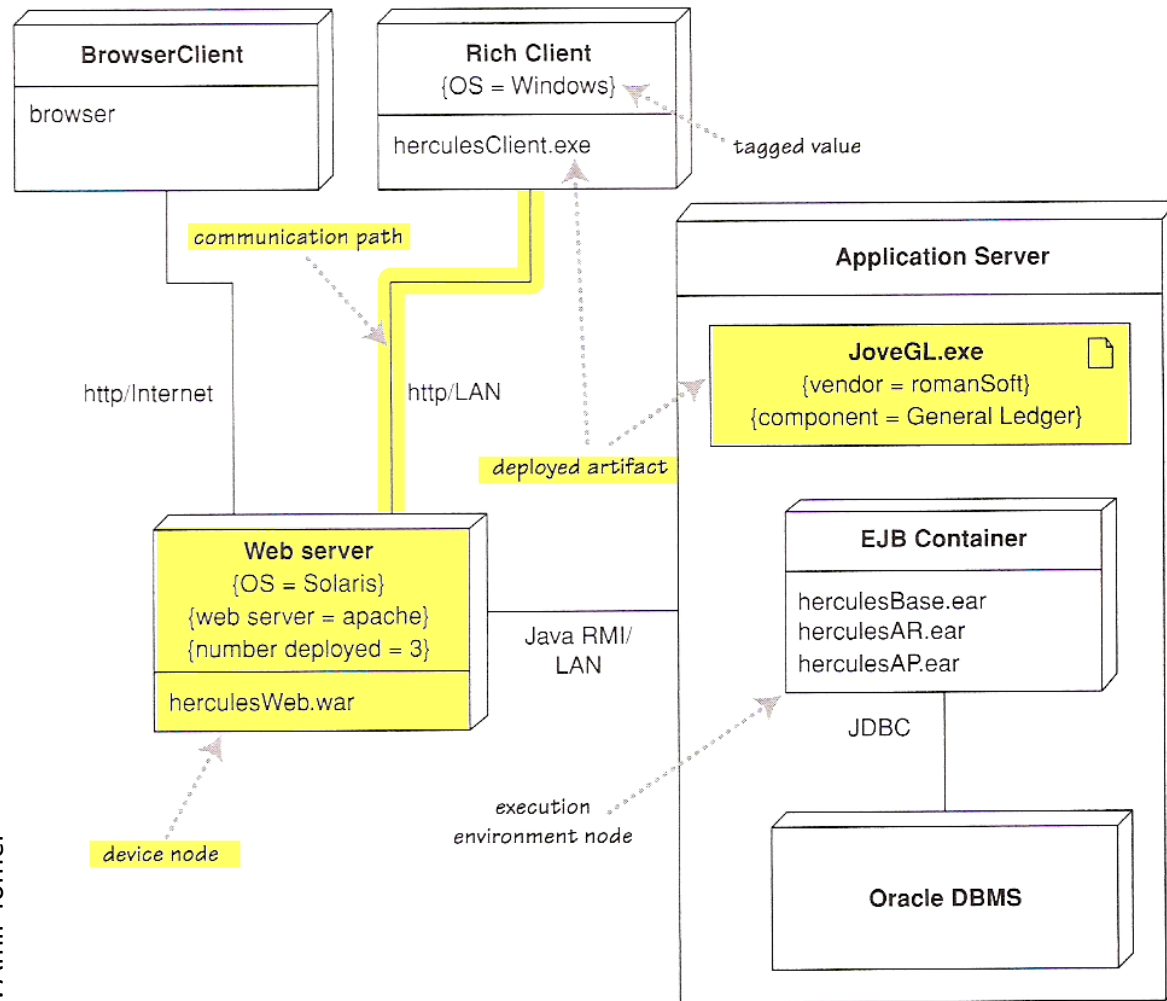
- תמונות
- קבצי אודיו / וידאו

- **פריטי מידע**

- קבצי עזרה
- מדריכים מקוונים



תרשים פריסה (Deployment Diagram)



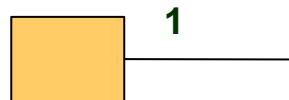
- תרשים המתאר את צמתי החומרה, הממשקים שביניהם ופריטי התוכנה הפרוסים בהם
- תרשים ארכיטקטורה מערכתית

– תרשים פריסה המציג את כל פריטי החומרה (HWCI) והתוכנה (CSCI) מהם מורכבת המערכת

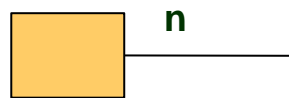
source: Fowler, M., *UML Distilled*, Addison-Wesley, 2003

ריבוי (multiplicity)

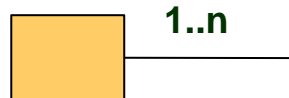
- תחום של שלמים לא שליליים, כולל אינסוף, המגדיר את מספר המופעים של ישות מסויימת.
- הריבוי מופיע בדרך כלל בקצה של קשר, המתאר יחס בין הישות לישות אחרת
- דוגמאות לריבוי:



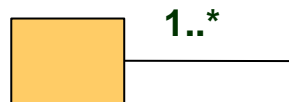
– פעם אחת בדיוק $[1,1]$



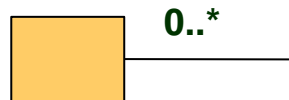
– n פעמים בדיוק $[n,n]$



– לא פחות מפעם אחת, ולא יותר מ-n פעמים $[1,n]$



– לא פחות מפעם אחת, אך ללא הגבלה $[1,\infty]$

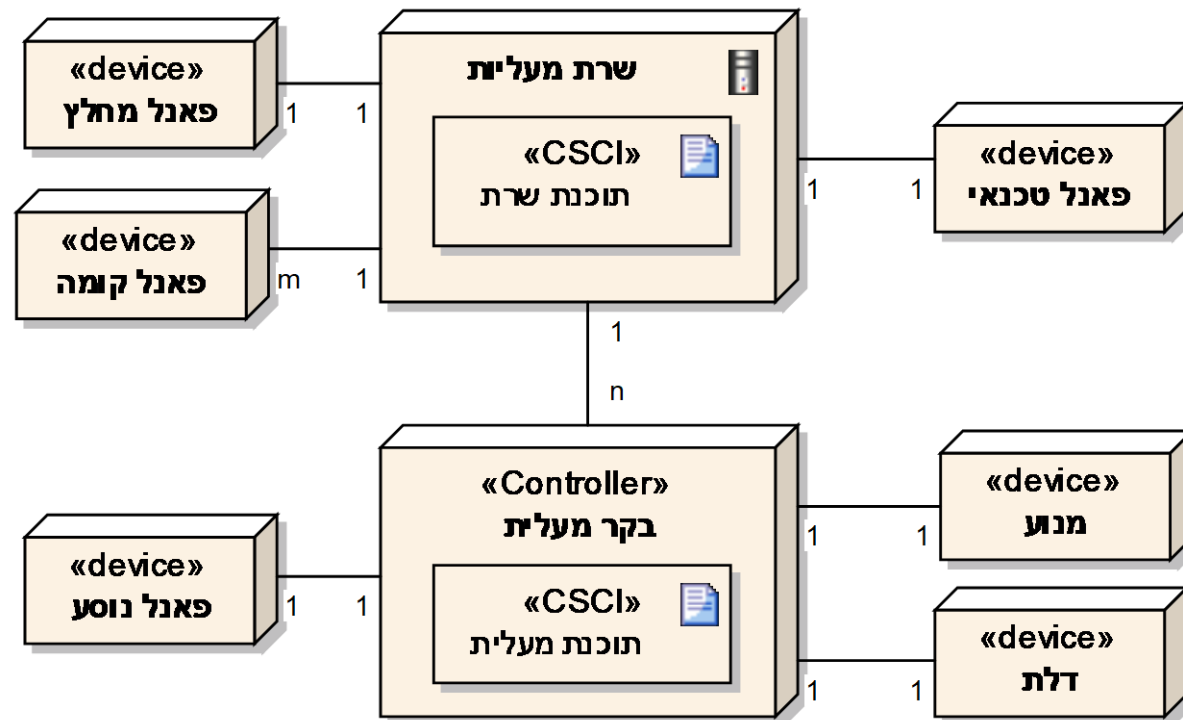


– מספר כלשהו של פעמים (כולל אפס) $[0..\infty]$

ארכיטקטורת מערכת המעליות – חלופה א'

- ארכיטקטורת שרת-לקוח

- כל מעלית היא צומת בעלת מיחשוב אוטונומי ושירותים מקומיים לנוסעיה
- שרת מרכזי מנהל ומבקר את כלל המערכת ונותן שירותים מרכזיים (אחזקה, חילוץ)
- קשר ישיר בין כל מעלית לשרת

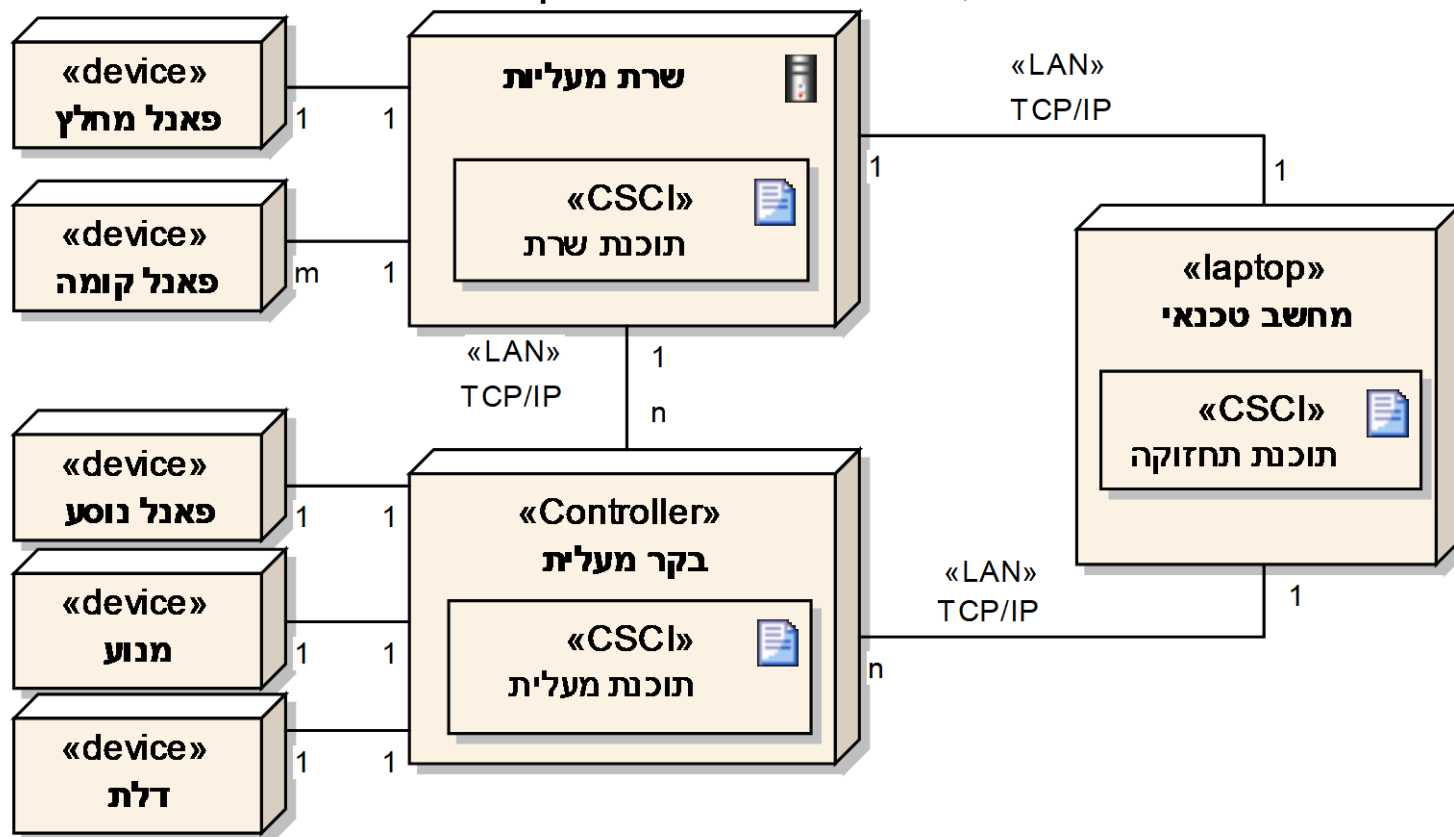


CSCI - Computer Software Configuration Item

ארכיטקטורת מערכת המעליות – חלופה ב'

- ארכיטקטורת רשת

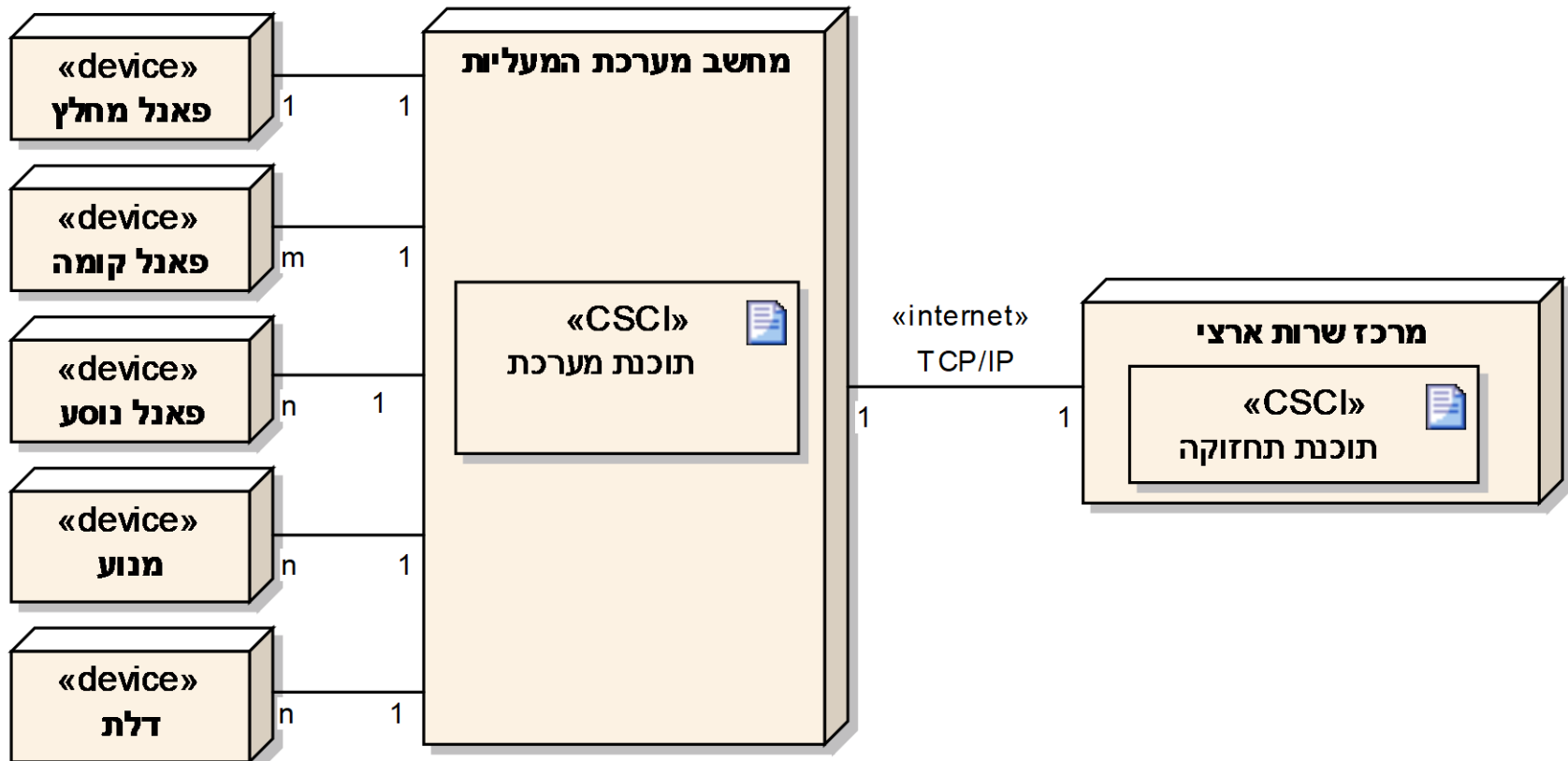
- השרת המרכזי והמעליות מחוברים ברשת תקשורת מקומית
- הטכנאי מגיע עם מחשב נייד, המתחבר למערכת דרך הרשת



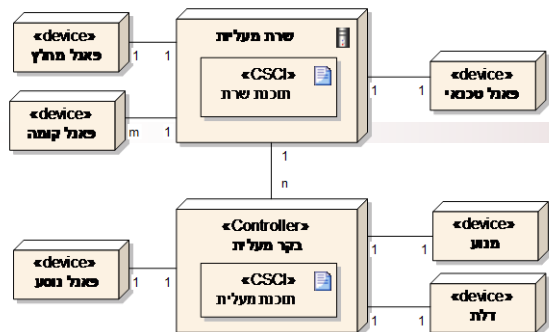
ארכיטקטורת מערכת המעליות – חלופה ג'

- ארכיטקטורה ריכוזית

- כל המערכת נשלטת ומתופעלת באמצעות מחשב אחד
- שרותים חיצוניים (בדיקה) ניתנים מרחוק, דרך רשת האינטרנט



טבלת ממשקים פיזיים (ארכיטקטורה א')



מרכיב	מרכיב	תכנים	תווך / פרוטוקול
שרת מעליות	בקר מעלית	< הקצאת נסיעות > סטטוס > קריאת מצוקה	חיבור ישיר
פאנל מחלץ	שרת מעליות	> קריאת חילוץ < פקודות חילוץ	חיבור ישיר
פאנל קומה	שרת מעליות	> הזמנות נסיעה < הדלקה/כיבוי כפתורי קומה	חיבור ישיר
פאנל טכנאי	שרת מעליות	< פקודות בדיקה > חיוויים וסטטוסים	חיבור ישיר
פאנל נוסע	בקר מעלית	< בקשות נסיעה > הדלקה/כיבוי כפתורי מעלית	חיבור ישיר
מנוע	בקר מעלית	> פקודות נסיעה/עצירה < סטטוס	חיבור ישיר
דלת	בקר מעלית	> פקודות פתיחה/סגירה < סטטוס	חיבור ישיר

- על בסיס סיפור הלקוח

- זהה צמתי חומרה

- זהה פריטי תוכנה

- ערוך Deployment Diagram

- ערוך טבלת ממשקים פיזיים

פריטי חומרה (HardWare Configuration Items - HWCI)

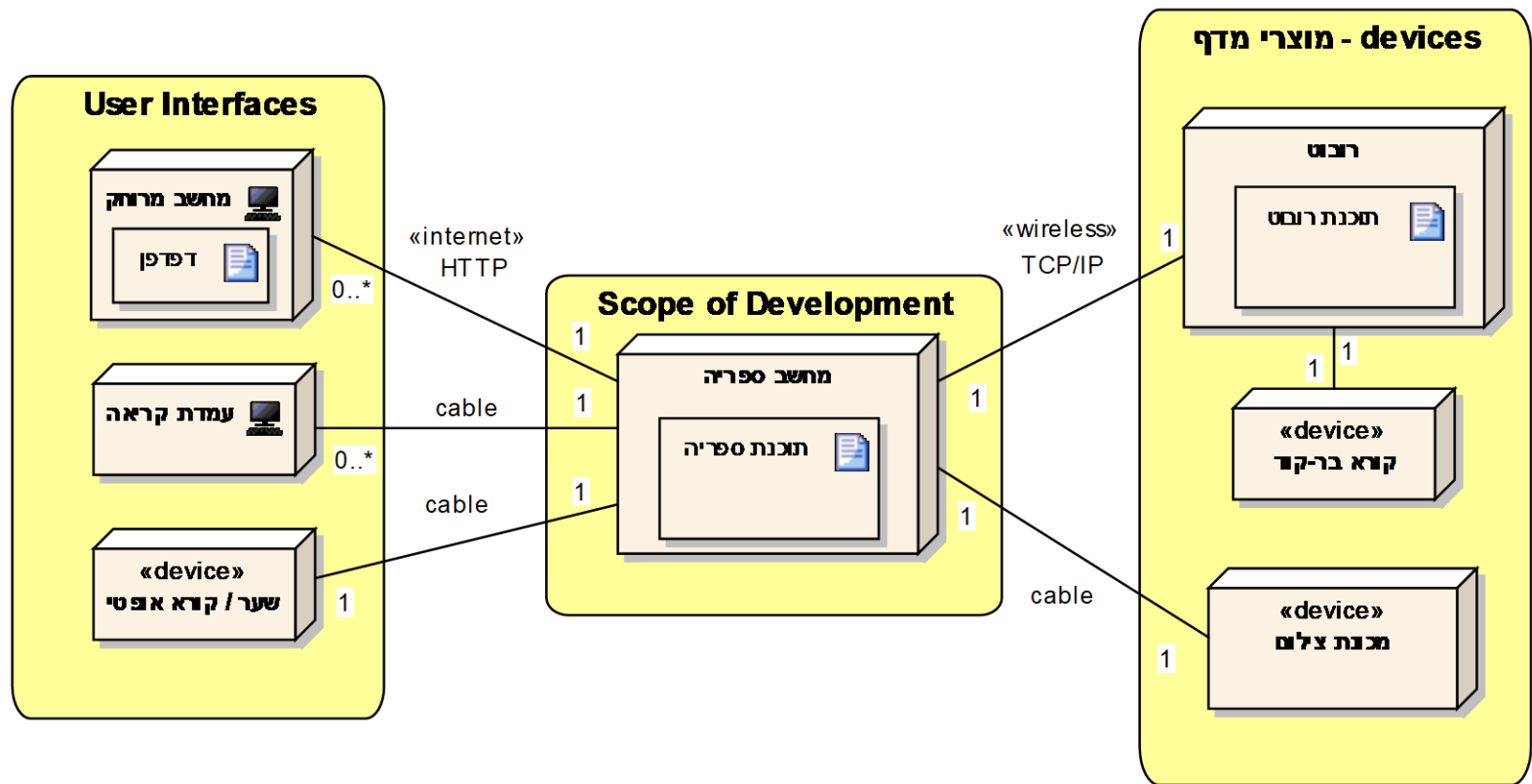
שם	מהות	דרישות נתמכות
רובוט	הרובוט המבצע את משימות השינוע בתחום הספריה	7
קורא ברקוד	אמצעי של הרובוט לזיהוי לקיחה ומסירה של ספרים	17
מחשב ספריה	המחשב המרכזי של RoboLib	2
מחשב מרוחק	מחשב כלשהו המשמש לגישה דרך האינטרנט	6
עמדת קריאה	עמדה בה משתמשים הקוראים לקבלת שירותי הספריה	2
שער	שער כניסה / יציאה לצורך זיהוי הוצאה / הכנסה של ספרים באמצעות קורא באופטי	46 ,44
מכונת צילום	מכונת צילום אוטומטית	37

פריטי תוכנה (Computer Software Configuration Items - CSCI)

שם	מהות	דרישות נתמכות
תוכנת ספריה	התוכנה המשמשת לניהול הספריה ומתן השרותים לקורא	1
תוכנת רובוט	התוכנה המשמשת להפעלת הרובוט לניהול המשימות	11, 12

System Deployment Diagram

deployment RoboLib System Architecture



ממשקים

מרכיב	מרכיב	תכנים	תווך / פרוטוקול	דרישות נתמכות
רובוט	מחשב	> הוראות שינוע	רשת אלחוטית /	8
	ספריה	< מיקום / סטטוס	TCP/IP	7
רובוט	קורא	> לקיחה / מסירה של עותק	חיבור ישיר	31 + 17
	בר-קוד			
מחשב	מחשב	< טפסי הרשמה	אינטרנט /	5
ספריה	מרוחק	> נתוני הרשמה	HTTP	
מחשב	עמדת	> נתוני הפעלה (סיסמה)	חיבור ישיר	18
ספריה	קריאה	> בקשות שרות		57, 21, 20, 19
		< קטלוג		24
		> < אינטראקציה עם קורא		56, 55, 54, 53
מחשב	מכונת	> סטטוס צילום	חיבור ישיר	56
ספריה	צילום			
מחשב	שער	> כניסת ספר	חיבור ישיר	45
ספריה		> יציאת ספר		46

ארכיטקטורת תוכנה

- **ארכיטקטורת תוכנה**

- חלוקת התוכנה לרכיבים

- הקצאת פונקציונאליות לכל רכיב

- הגדרת ממשקים בין הרכיבים

- **שמירת עקביות (consistency) בין ארכיטקטורת התוכנה לארכיטקטורת המערכת**

- התוכנה מותקנת על גבי החומרה

- ממשקי התוכנה מתממשים באמצעות ה"פלטפורמה" של ממשקי החומרה

- **חייבת להיות התאמה ברורה בין ממשקי התוכנה לממשקי החומרה!**

מהו רכיב תוכנה

- תוכנה (פריט/יישום) בנויה מרכיבי תוכנה

- רכיב תוכנה (software component)

- חלק מודולרי של התוכנה המסתיר את מימושו מאחורי קבוצת ממשקים חיצוניים

- בתוך מערכת, רכיבים המספקים ממשקים זהים ניתנים להחלפה זה בזה באופן חופשי

- רכיבים יכולים לבוא ממקורות שונים

- פיתוח עצמי

- שימוש חוזר

- קוד פתוח (open source)

- רכיבי מדף

פונקציונאליות (Functionality)

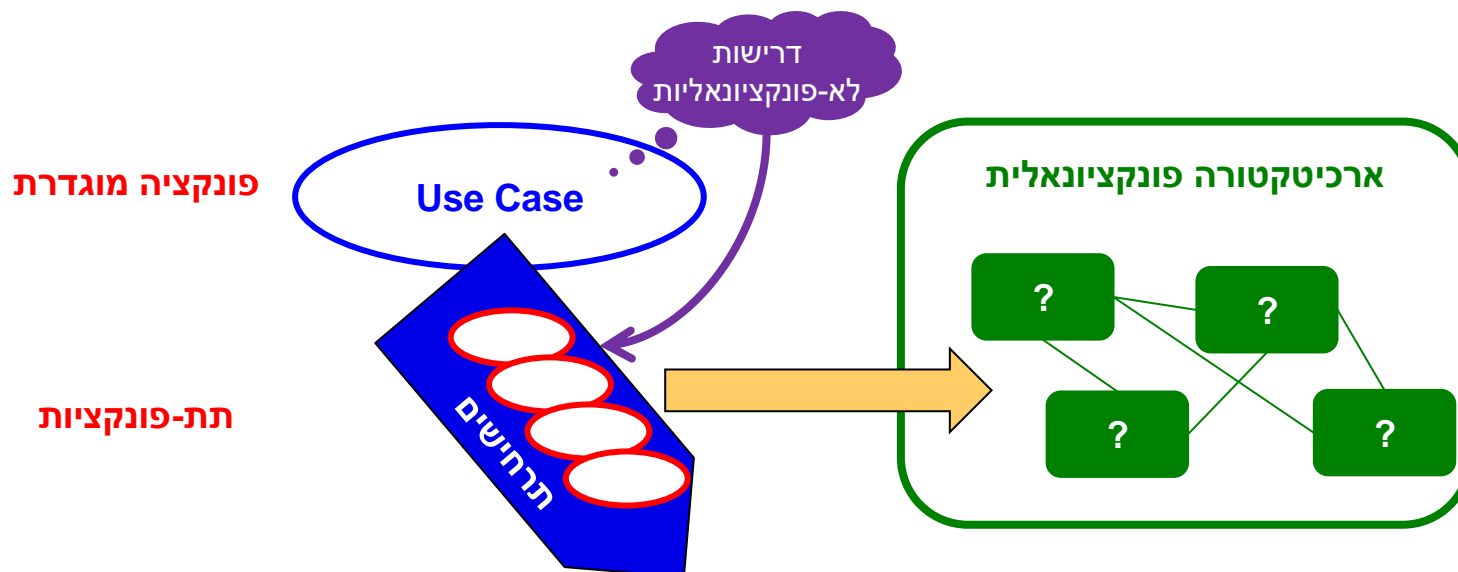
- **פונקציונאליות** היא יכולת המערכת לעשות את העבודה שלשמה היא נועדה
- הקשר בין פונקציונאליות לארכיטקטורה
 - פונקציונאליות איננה קובעת את הארכיטקטורה
 - בהינתן פונקציונאליות נדרשת, אין גבול למספר הארכיטקטורות שניתן ליצור כדי לספק פונקציונאליות זו
 - הפונקציונאליות מגדירה בעיקר את אוסף היכולות הנדרשות **ממרכיבי** הארכיטקטורה
 - ומצד שני, מאפייני איכות (דרישות לא-פונקציונאליות) כשלעצמם אינם מגדירים מערכת
 - הלקוח לא מבקש "מערכת אמינה ובטוחה" אלא "מערכת שעושה <משהו> באופן אמין ובטוח"
 - פונקציונאליות ומאפייני איכות הם אורתוגונאליים!
 - אבל הטיפול במאפייני איכות עשוי לדרוש פונקציונאליות נוספת מהמערכת
- מודול זה מוקדש לניתוח הפונקציונאלי ולהגדרת ארכיטקטורה פונקציונאלית
- המודול הבא יוקדש להתאמת הארכיטקטורה למאפייני האיכות הנדרשים

תרחישים פונקציונאליים

- הפונקציונאליות של מערכת מושגת באמצעות אוסף של פונקציות (יכולות, features), הנמצאות במרכיבים שונים של המערכת
- תרחיש פונקציונאלי הוא הפעלת **רצף של פונקציות** לצורך השגת תוצאה או מילוי משימה כלשהי
- מודל Use Case מגדיר את אוסף התרחישים הפונקציונאליים של המערכת
 - תרחיש ההצלחה הראשי
 - הסתעפויות
- למרות שזיהינו דרישות לא-פונקציונאליות והצמדנו אותם ל-Use Cases עדיין לא טיפלנו בהם
 - במודול הבא נראה כיצד אפשר לממש עמידה במאפייני איכות ע"י הפיכתם לתרחישים פונקציונאליים

ניתוח פונקציונאלי (Functional Analysis*)

- בחינה של פונקציה מוגדרת, לצורך
 - זיהוי תת-פונקציות הנדרשות לצורך מימוש פונקציה זו
 - זיהוי יחסים וממשקים פונקציונאליים (חיצוניים ופנימיים) ובניית ארכיטקטורה פונקציונאלית
 - "הורדת" דרישות ביצועים [ומאפייני איכות אחרים] מפונקציית-העל והקצאתן לתת-פונקציות ברמה הנמוכה יותר



*ISO/IEC/IEEE 24765:2010 Systems and software engineering—Vocabulary

זיהוי הפונקציות בתרחישי Use Cases – דוגמה

SUC-2	נסיעה במעלית
Trigger	• הנוסע נכנס למעלית*
MSS	<ol style="list-style-type: none"> 1. הנוסע לוחץ על כפתור הקומה המבוקשת והבקשה נרשמת 2. הכפתור שנלחץ נדלק 3. דלת המעלית נסגרת (במידה והיתה פתוחה) 4. המעלית ממשיכה בנסיעה באותו כיוון, אם קיימות בקשות קודמות 5. המעלית נעצרת בקומה הבאה אליה קיימת בקשה 6. הדלת נפתחת 7. כפתור הקומה כבה 8. חזרה לצעד 3, כל עוד לא הגיע הנוסע לקומת היעד 9. הנוסע יוצא מהמעלית*
הסתעפות א'	<p>חלופה מצעד 4 של MSS: אין בקשות באותו כיוון / המעלית הגיעה לקצה המסלול</p> <ol style="list-style-type: none"> 1א4. המעלית מחליפה את כיוון הנסיעה 2א4. התרחיש נמשך
הסתעפות ב'	<p>חריגה בצעד 4 של MSS: הנוסע לחץ על כפתור עצירת חירום</p> <ol style="list-style-type: none"> 1ב4. המעלית נעצרת מיד 2ב4. המעלית מבטלת את כל בקשות העצירה הקיימות 2ב4. התרחיש מסתיים
הסתעפות ג'	<p>חריגה בצעד 4 של MSS: המעלית נתקעה</p> <ol style="list-style-type: none"> 1ג4. הנוסע מזעיק חילוץ 2ג4. מתחיל חילוץ [extending UC-3]

ארכיטקטורה פונקציונאלית

- הארכיטקטורה הפונקציונאלית מורכבת מרכיבים לוגיים (פונקציונאליים) ומהקשרים ביניהם
 - בקרה וסדר פעולות
 - העברת מידע
- רכיב לוגי (פונקציונאלי) יכול להיות כל ישות אשר
 - יכולה לבצע פונקציה אחת או יותר
 - יש לה ממשק אחד או יותר דרכו יכולים רכיבים אחרים, או ישויות חיצוניות, ליזום את ביצוע הפונקציה, עם או בלי העברת מידע אל ומאת הרכיב
 - יכולה לקבל שירותים (ביצוע פונקציות) מרכיבים אחרים או מישויות חיצוניות
 - **איננה תלויה מימוש** (כלומר, ניתן לממש את הרכיב באופנים שונים)

ארכיטקטורה פונקציונאלית - המשך

- כל תרחיש בכל UC עשוי לתאר סדר הפעלה שונה ומעבר מידע שונה בין הפונקציות
 - ולכן הארכיטקטורה הפונקציונאלית צריכה לשקף את עקרונות ההפעלה והעברת המידע שיוכלו לאפשר את כל התרחישים
- גישה מקובלת במערכות עתירות-תוכנה: חלוקה לרבדים
 - רובד (tier) הוא ריכוז של רכיבים לוגיים בעלי פונקציונאליות דומה

שימו לב

אנו משתמשים כאן במונח רובד (tier) ולא במונח שכבה (layer), למרות שבמקומות רבים אין הבחנה בין המושגים. את מושג השכבות נגדיר בהמשך.

רבדים פונקציונאליים – Functional Tiers

- **רבדים מקובלים במערכות עתירות תוכנה**

- רובד הנתונים (data tier)

- רכיבים האחראיים לאיחסון ולאיחזור נתונים

- רובד היישומים (application tier)

- רכיבים האחראיים לפונקציות העיבוד השונות במערכת

- רובד התצוגה (presentation tier)

- רכיבים של ממשקי-משתמש האחראים לאינטראקציה עם עמדות ההפעלה

- רובד הבקרה (control tier)

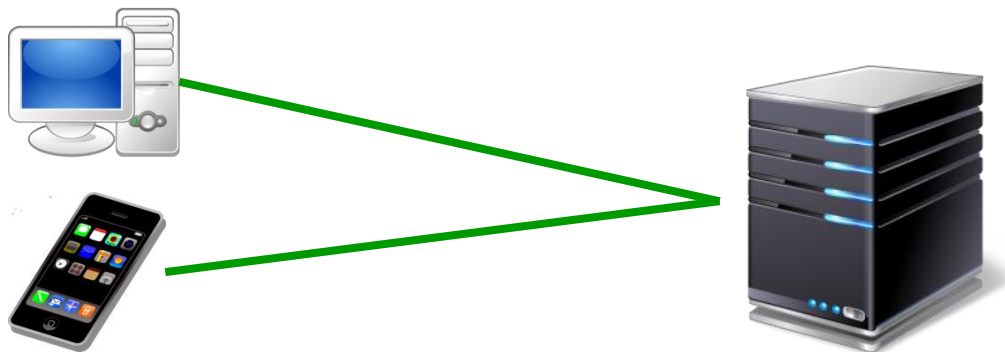
- רכיבים האחראיים לבקרה על התקנים חיצוניים ("דרייברים")

- **את הרבדים הפונקציונאליים ניתן למקם בצמתים שונים בארכיטקטורה הפיזית**

- ר' דוגמאות בשקף הבא

הקצאת רבדים פונקציונאליים לצמתים פיזיים

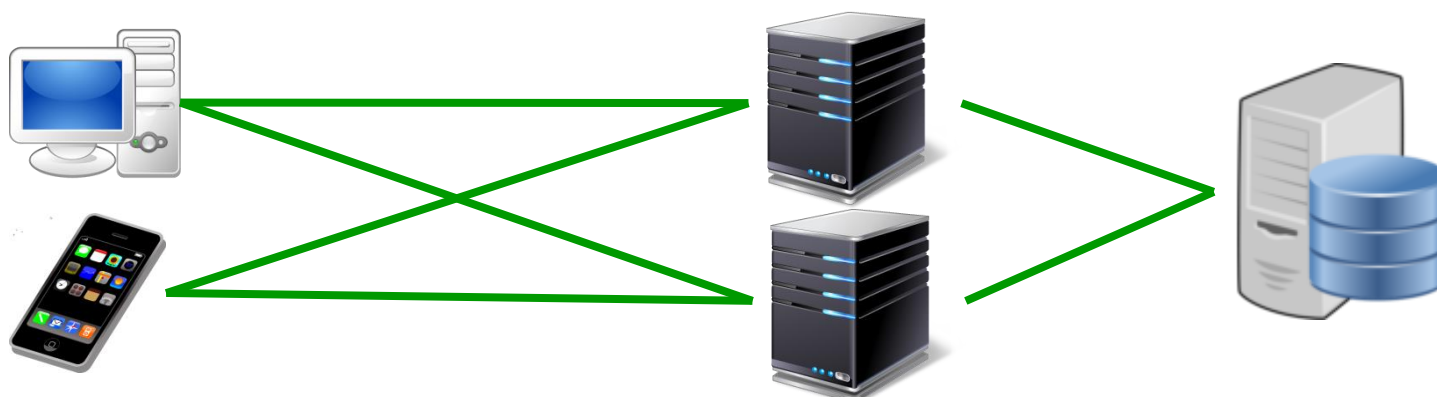
- client-server architecture



Clients (presentation + application)

Server (application + data)

- 3-tier architecture



Clients (presentation)

Application Servers (application)

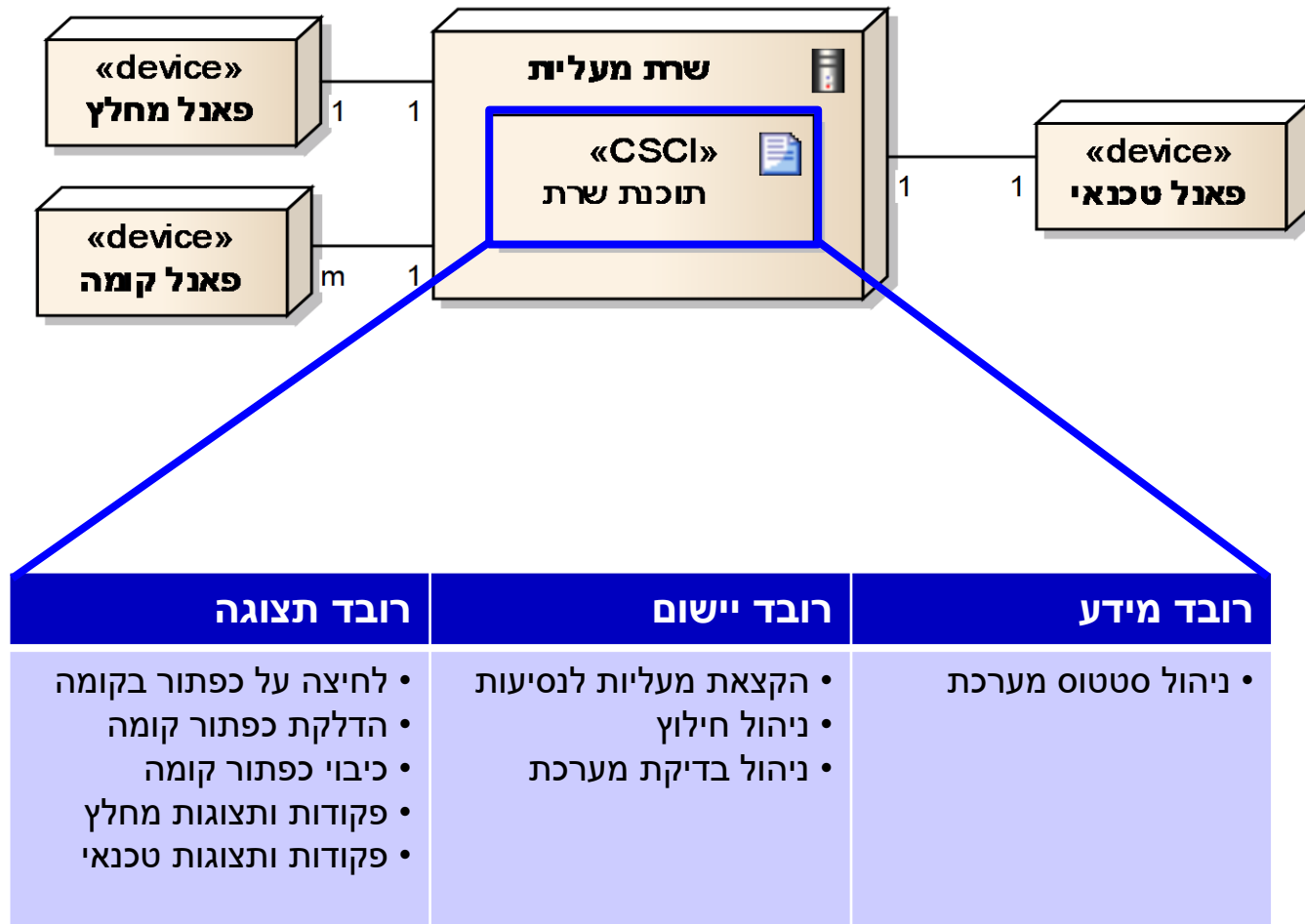
Data Server (data)

חלוקת הפונקציות של תוכנת המעלית בין רבדים



רובד בקרה	רובד מידע	רובד יישום	רובד תצוגה
<ul style="list-style-type: none"> • הסעת מעלית • עצירת מעלית • פתיחת דלת • סגירת דלת 	<ul style="list-style-type: none"> • רישום בקשה • בדיקת בקשות קיימות • ביטול בקשות 	<ul style="list-style-type: none"> • המשך נסיעה באותו כיוון • החלפת כיוון נסיעה 	<ul style="list-style-type: none"> • לחיצה על כפתור במעלית • הדלקת כפתור מעלית • כיבוי כפתור מעלית • בקשת עצירה • אזעקת חילוץ

חלוקת הפונקציות של תוכנת השרת בין רבדים



יתרונות החלוקה לרבדים

- לכידות הדוקה (tight cohesion)

- לכידות (cohesion)* : The manner and degree to which the tasks performed by a **single** software module are **related** to one another

– לכל רובד יש ייחודיות פונקציונאלית ברורה וניתן להקצות בקלות כל פונקציה לרובד המתאים

- צימוד רופף (loose coupling)

- צימוד (Coupling)*: The manner and degree of **interdependence** between software modules

– המימוש של רובד אחד איננו תלוי במימוש של רובד אחר, ולכן ניתן לשנות מימוש של רובד מבלי לשנות את מימוש הרבדים האחרים

*ISO/IEC/IEEE 24765:2010 Systems and software engineering—Vocabulary

רכיבים פונקציונאליים (Functional Components)

- **רכיב פונקציונאלי (רכיב לוגי, רכיב תוכנה) יכול להיות כל ישות אשר**

- יכולה לבצע פונקציה אחת או יותר [לספק שירותים]
- יש לה ממשק אחד או יותר דרכם יכולים רכיבים אחרים, או ישויות חיצוניות, ליזום את ביצוע הפונקציות [את קבלת השירותים], עם או בלי העברת מידע אל ומאת הרכיב

- יכולה ליזום ביצוע פונקציות [לקבל שירותים] מרכיבים אחרים או מישויות חיצוניות

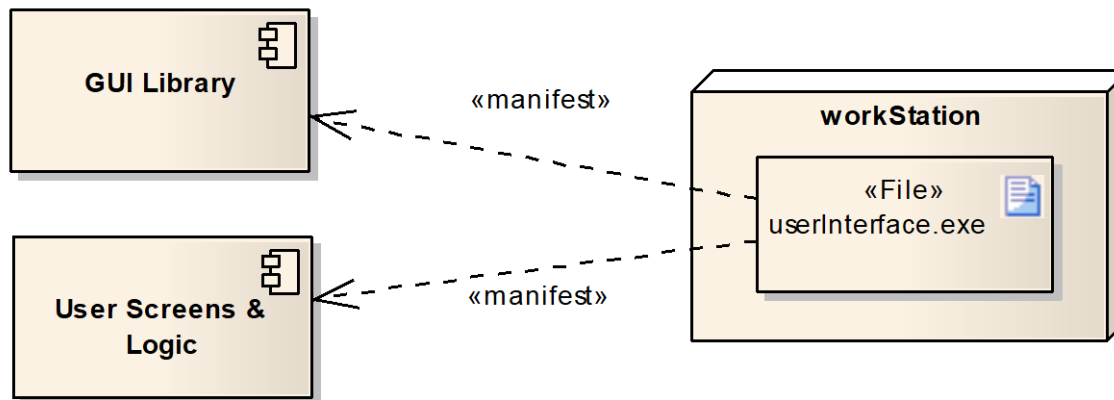
- **איננה תלוית מימוש (כלומר, ניתן לממש את הרכיב באופנים שונים)**

- **רכיב הינו "קופסה שחורה" מבחינה פונקציונאלית במובנים הבאים:**

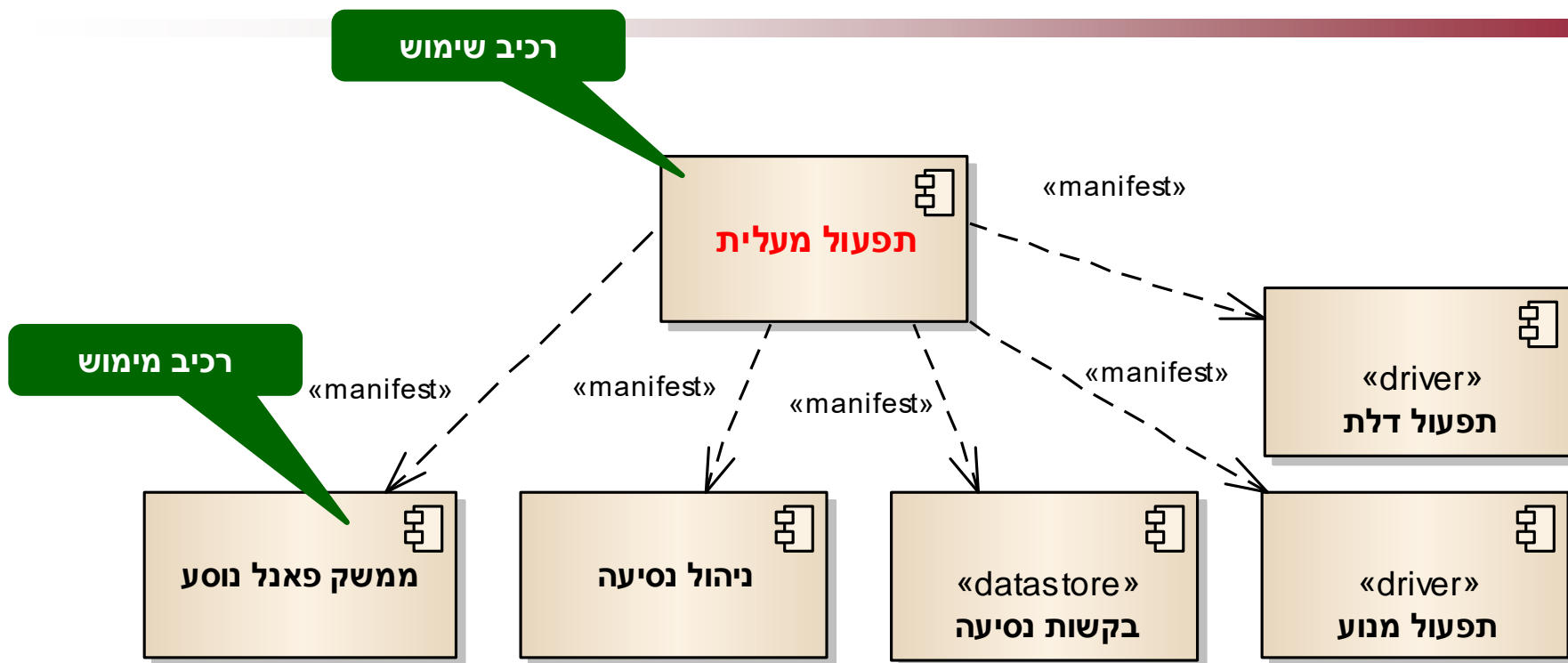
- ניתן לבחון את נכונותו הפונקציונאלית רק באמצעות הקלט והפלט
- ניתן למדוד את ביצועיו (כגון מהירות תגובה) רק במדידה מבחוץ
- ניתן להחליפו ברכיבים הזוהים אליו מבחינה פונקציונאלית, ללא שינוי בממשקיו

רכיבי שימוש לעומת רכיבי מימוש

- **רכיבי שימוש / פריטי תוכנה (software artifacts)**
 - רכיבים/פריטים המותקנים בסביבת הריצה ונחוצים לפעולה תקינה של המערכת
- **רכיבי מימוש (software components)**
 - המרכיבים מהם יוצרים את התוכנה בסביבת הפיתוח
 - חלקי קוד
 - ספריות ותוכנות מדף
- **פריטי תוכנה משקפים (<<manifest>>) רכיבי תוכנה**
 - בשלב אינטגרציית התוכנה (שילוב רכיבים ובדיקת פריטים) מרכיבים את פריטי התוכנה מהרכיבים

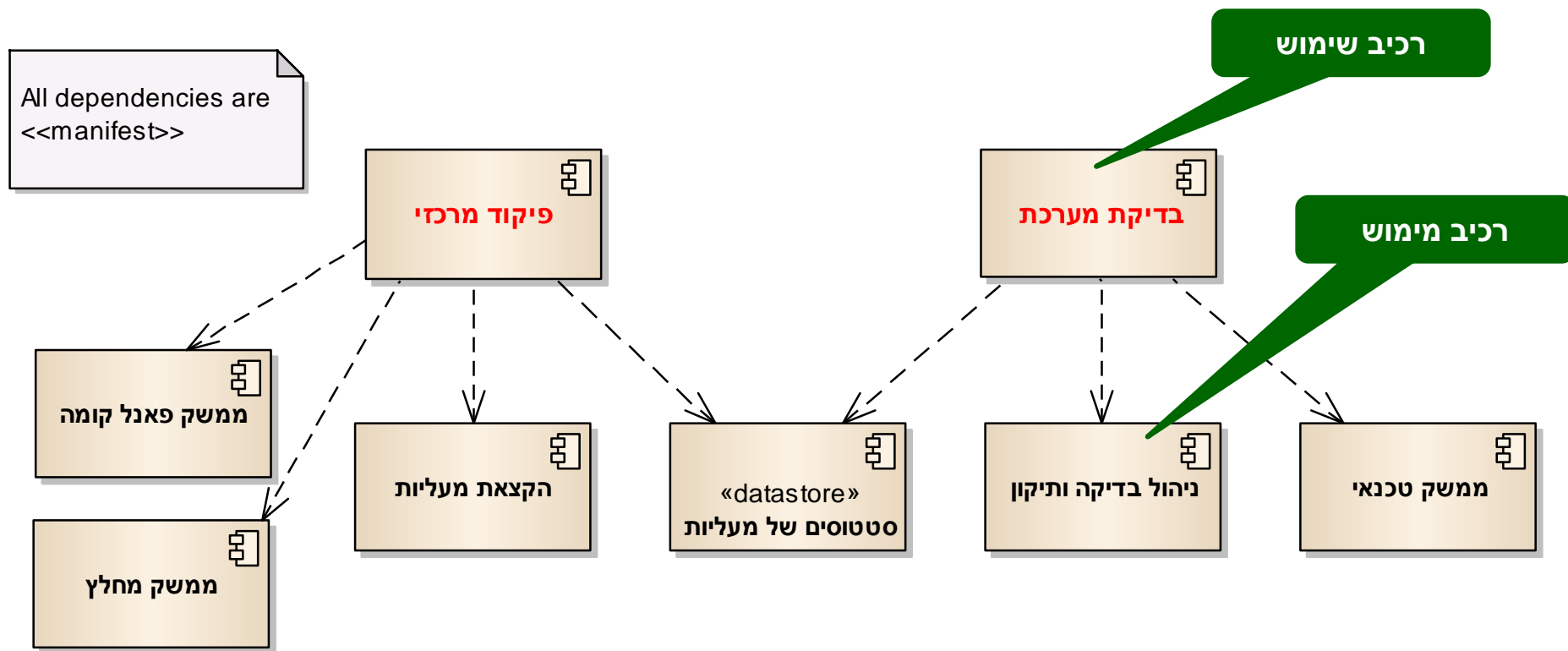


תוכנת המעלית – חלוקה אפשרית לרכיבים



רובד בקרה	רובד מידע	רובד יישום	רובד תצוגה
<ul style="list-style-type: none"> פתיחת דלת סגירת דלת הסעת מעלית עצירת מעלית 	<ul style="list-style-type: none"> רישום בקשה בדיקת בקשות קיימות ביטול בקשות 	<ul style="list-style-type: none"> המשך נסיעה באותו כיוון החלפת כיוון נסיעה 	<ul style="list-style-type: none"> לחיצה על כפתור במעלית הדלקת כפתור מעלית כיבוי כפתור מעלית בקשת עצירה אזעקת חילוץ

תוכנת השרת - חלוקה אפשרית לרכיבים

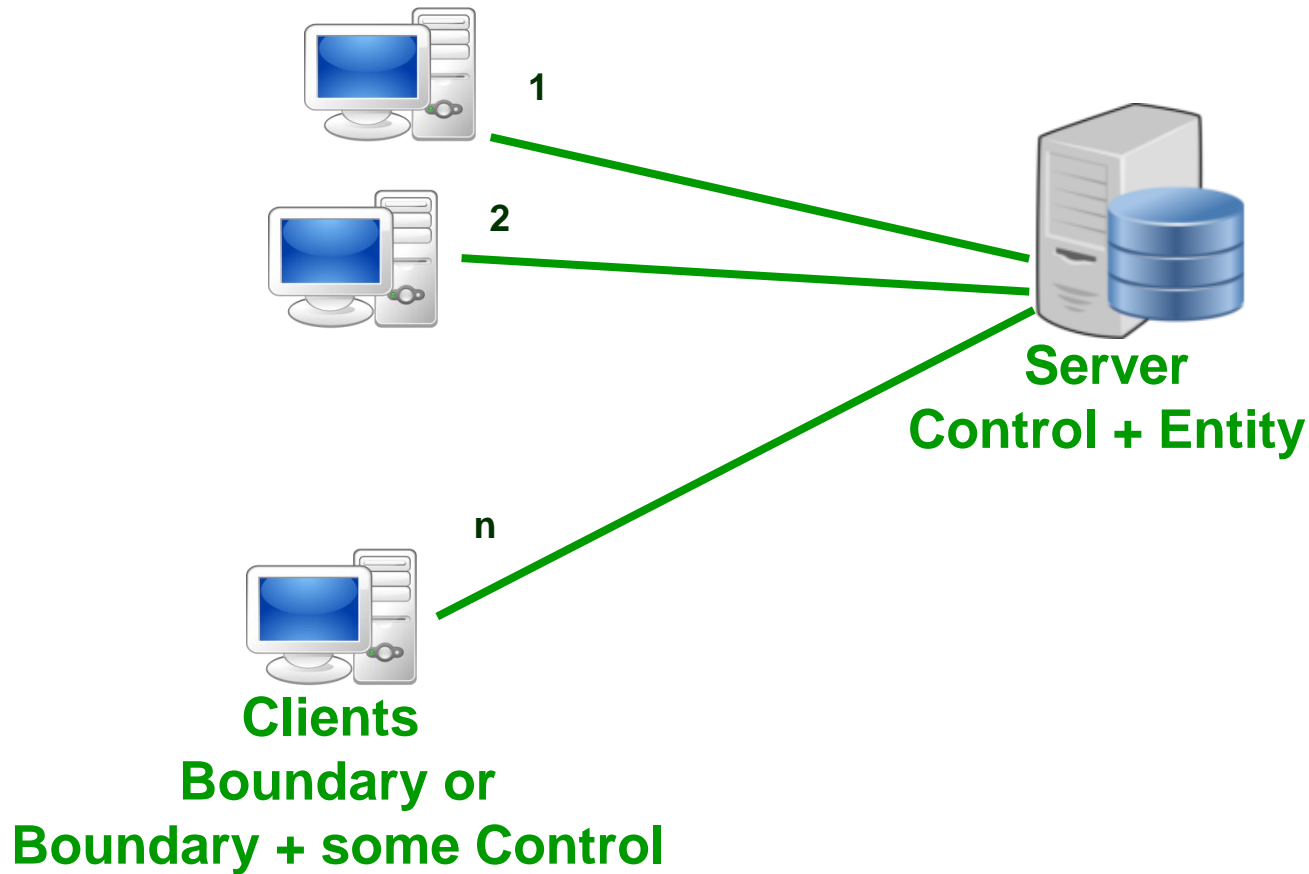


רובד תצוגה	רובד יישום	רובד מידע	רובד יישום	רובד תצוגה
<ul style="list-style-type: none"> • לחיצה על כפתור בקומה • הדלקת כפתור קומה • כיבוי כפתור קומה • פקודות ותצוגות מחלץ 	<ul style="list-style-type: none"> • הקצאת מעליות לנסיעות • ניהול חילוץ 	<ul style="list-style-type: none"> • ניהול סטטוס מערכת 	<ul style="list-style-type: none"> • ניהול בדיקת מערכת 	<ul style="list-style-type: none"> • פקודות ותצוגות טכנאי

מה קורה במערכת שיש לפתח בקורס?

- הדרישה היא לפתח מערכת בארכיטקטורה של שרת – לקוח ושל שלושת הרבדים (3-tier).
- מודל שלושת הרבדים כולל את הרבדים הבאים:
 - רובד הנתונים (data tier):
 - רכיבים האחראיים לאיחסון ולאיחזור נתונים.
 - הרכיבים נקראים גם **רכיבי ישות - Entity**
 - רובד היישומים (application tier)
 - רכיבים האחראיים לפונקציות העיבוד השונות במערכת
 - הרכיבים נקראים גם **רכיבי בקרה - Control**
 - רובד התצוגה (presentation tier)
 - רכיבים של ממשקי-משתמש האחראים לאינטראקציה עם עמדות ההפעלה
 - הרכיבים נקראים גם **רכיבי ממשק – Boundary**

הארכיטקטורה של המערכת שיש לפתח בקורס



ממשקים פונקציונליים

- רכיב מקיים אינטראקציה עם סביבתו (רכיבים אחרים או ישויות חיצוניות) באמצעות שתי קבוצות של ממשקים:

- ממשקים מסופקים (Provided Interfaces)

– קרי: ממשקים לשירותים **המסופקים** ע"י הרכיב

- האמצעים שחושף הרכיב לרשות סביבתו לצורך קבלת השירותים ממנו
- דוגמאות

– פונקציות ציבוריות (שם, פרמטרים וסוגיהם, סוג ערך מוחזר)

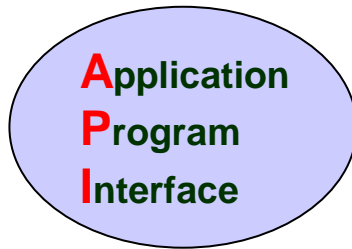
– ממשק לשאילתות SQL

– קליטת קוד הפעלה דרך עינית

– קבלת קובץ

– Dialog box

– טופס HTML



- ממשקים נדרשים (Required Interfaces)

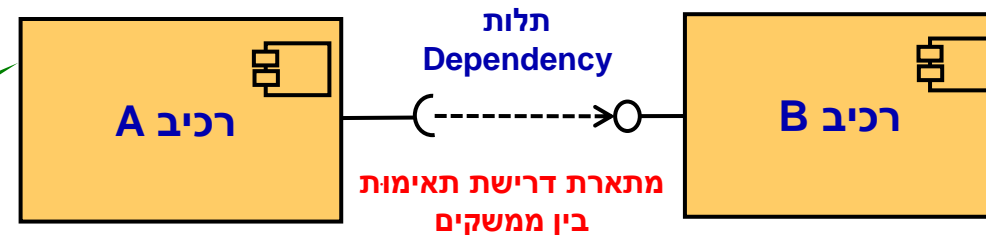
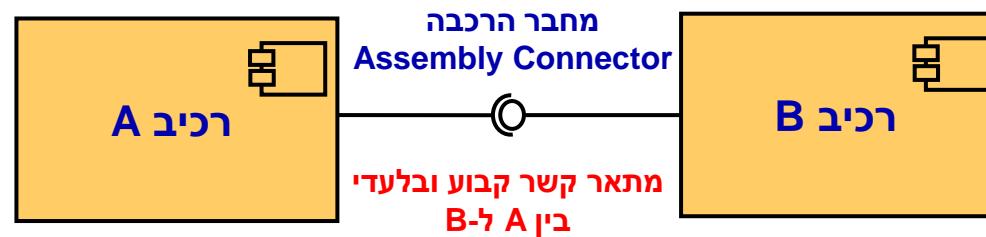
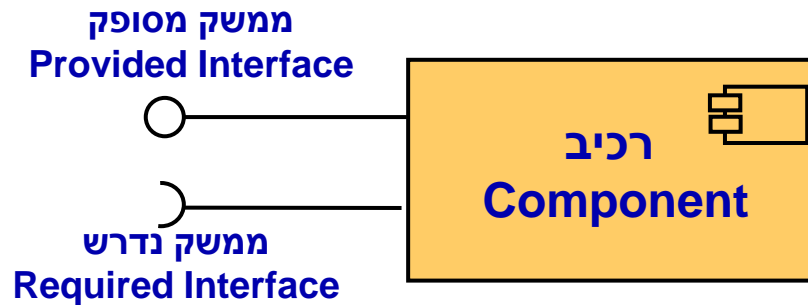
– קרי: ממשקים לשירותים **הנדרשים** ע"י הרכיב

- פניות של הרכיב לרכיבים אחרים דרך הממשקים המסופקים שלהם

- דוגמאות

– אותן דוגמאות, מהצד השני (פרט לממשקי HMI - Human Machine Interface)

רכיבים וממשקים ב-UML



A תלוי ב-B פירושו:
שינוי ב-B עלול
לדרוש שינוי ב-A

האם תמיד הממשק
הנדרש תלוי בממשק
המסופק?

מערכת המעליות – רכיב תפעול מעלית

- **שירותים (פונקציות)**

- זיהוי וניתוב בקשות נסיעה, עצירה, חילוץ
- ניהול משימות המעלית
- ניהול נסיעת המעלית

- **ממשקים מסופקים**

- בקשות מכפתורים
 - ממשק לקליטת בקשות נסיעה המגיעות מכפתורי המעלית
- בקשות ישירות
 - ממשק לקליטת בקשות נסיעה באופן ישיר (שלא באמצעות כפתורים)
- סטטוס
 - ממשק להספקת קוד סטטוס על פי בקשה

- **ממשקים נדרשים**

- נסיעה/עצירה
 - ממשק למנגנון ההנעה (המנוע) של מעלית פיזית
- סגירת/פתיחת דלת
 - ממשק למנגנון דלת פיזית



מערכת המעליות – רכיב פיקוד מרכזי

- **שירותים (פונקציות)**
 - זיהוי וניתוב בקשות עצירה
 - הקצאת מעליות
 - ניהול פעולות חילוץ
- **ממשקים מסופקים**
 - הזמנות מכפתורים
 - ממשק לקליטת הזמנות נסיעה המגיעות מכפתורי קומות
 - הזמנת מעליות לקומות
 - ממשק לקליטת הזמנות נסיעה באופן ישיר (שלא באמצעות כפתורים)
 - פקודות חילוץ
 - ממשק לקליטת פקודות הפעלה של מחלץ באמצעות מפתח/כפתורים יעודיים
- **ממשקים נדרשים**
 - בקשות נסיעה
 - הפניית בקשות נסיעה למעלית
 - סטטוס מעלית
 - קבלת סטטוס מעלית על פי דרישה

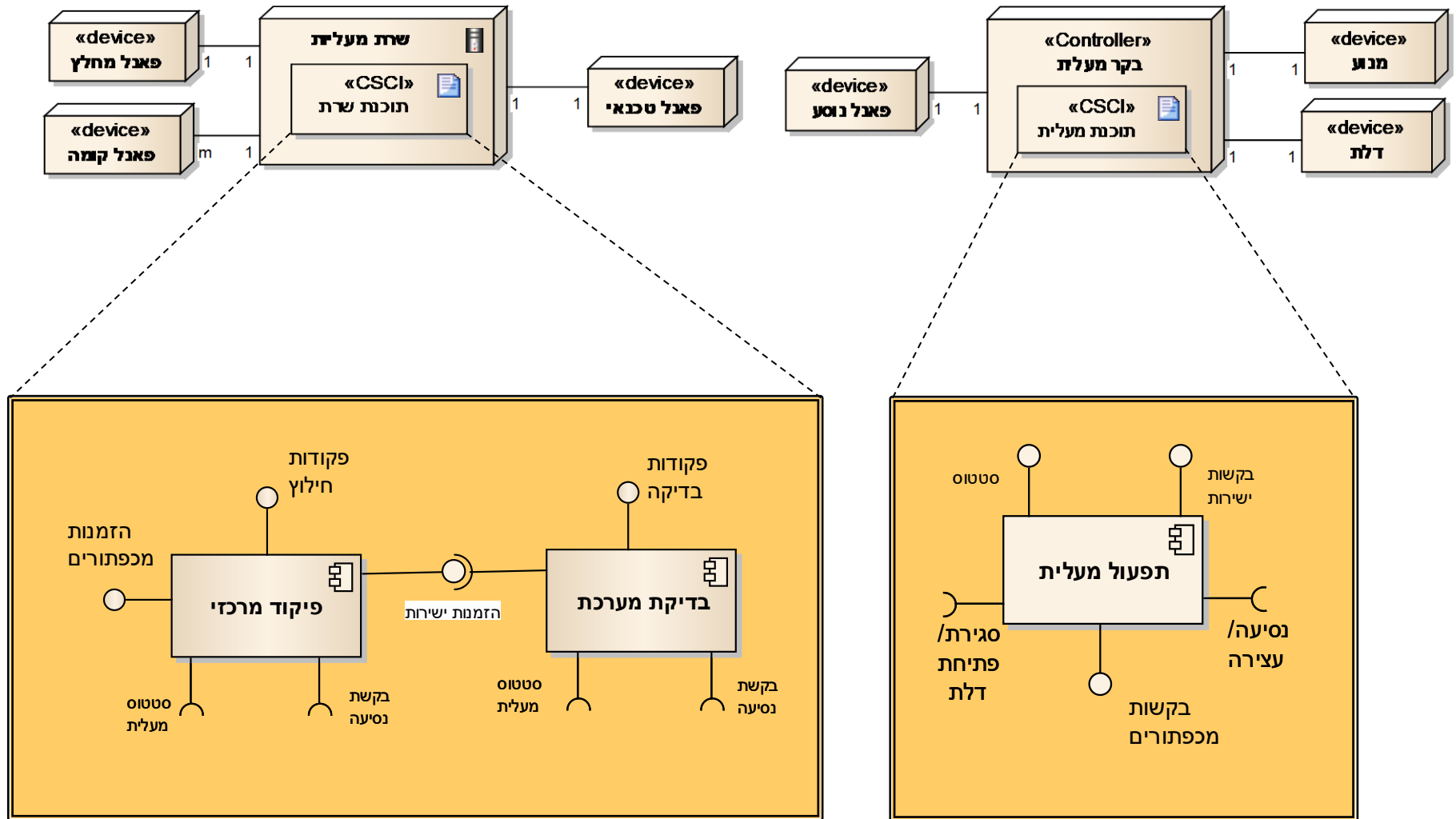


מערכת המעליות – רכיב בדיקת מערכת

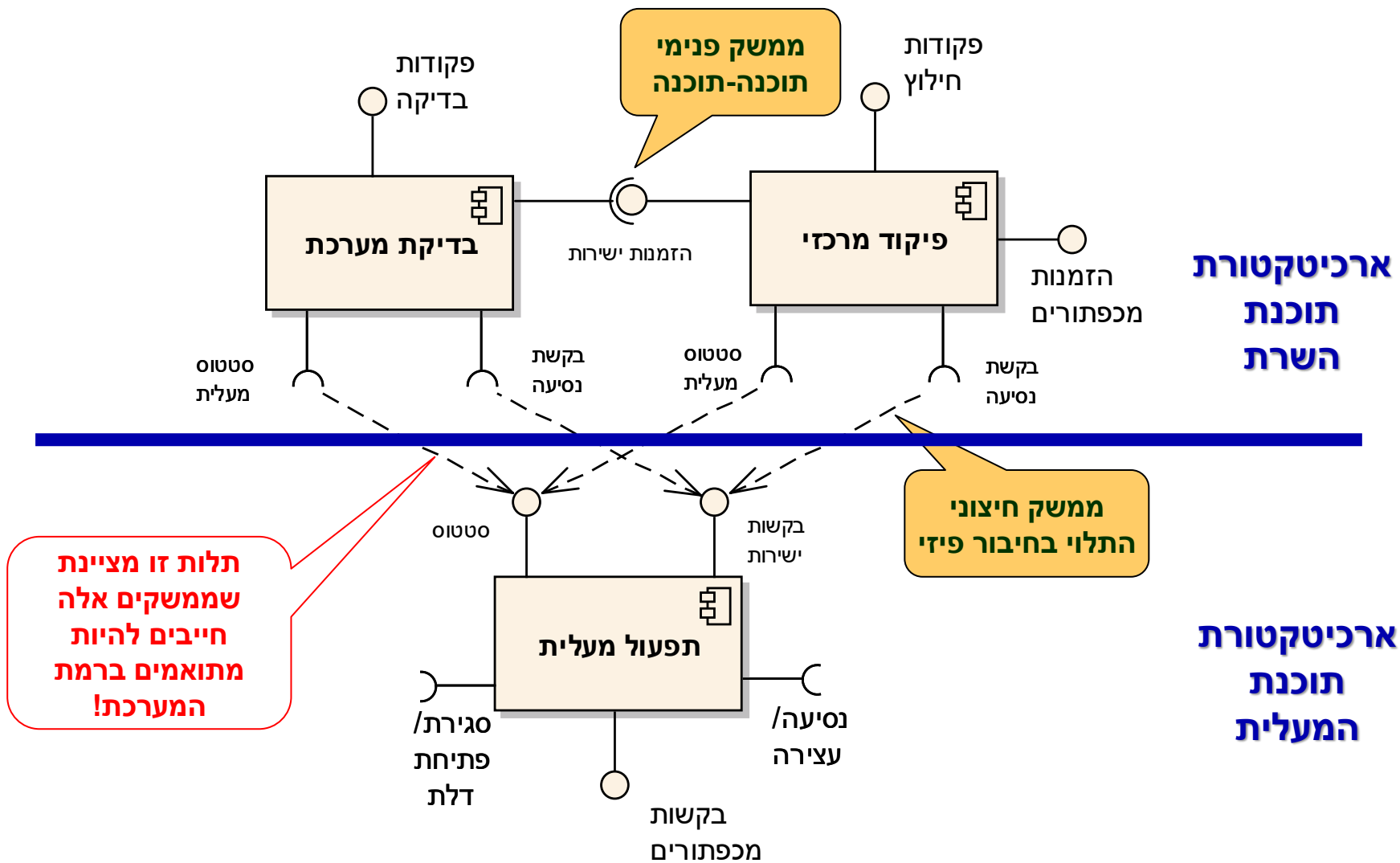
- **שירותים (פונקציות)**
 - זיהוי וניתוב פקודות טכנאי
 - ניהול בדיקה ותיקון של המערכת
- **ממשקים מסופקים**
 - פקודות בדיקה
 - ממשק-משתמש לטכנאי
- **ממשקים נדרשים**
 - הזמנת מעליות לקומות
 - הפניית הזמנות נסיעה לקומות
 - בקשות נסיעה
 - הפניית בקשות נסיעה למעלית
 - סטטוס מעלית
 - קבלת סטטוס מעלית על פי דרישה



תרשים רכיבים (Component Diagram): מבנה (ארכיטקטורה) של פריטי התוכנה



תוכנת מערכת המעלות – תרשים רכיבים משותף



תרשים רכיבים (Component Diagram)

- תרשים רכיבים הוא תרשים המציג

- רכיבים פונקציונאליים/לוגיים
- ממשקיהם המסופקים והנדרשים
- אופי הקשר בין הממשקים

- ארכיטקטורת התוכנה יכולה להכיל מספר תרשימי רכיבים

- תצורות שימוש (ריצה) שונות

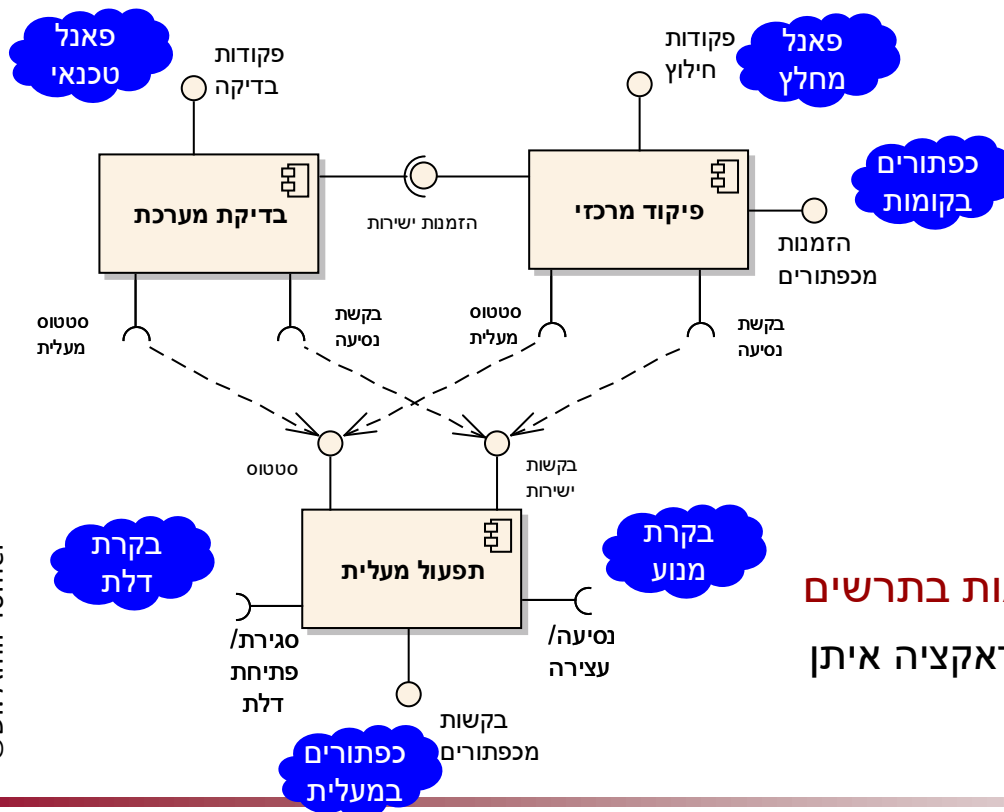
- תצורה מבצעית
- תצורת תחזוקה
- תצורת בדיקות

- תצורות מימוש (builds) שונות

- שימוש בספריות שונות
- שימוש ברכיבי מדף שונים
- פיתוח בקבוצות שונות

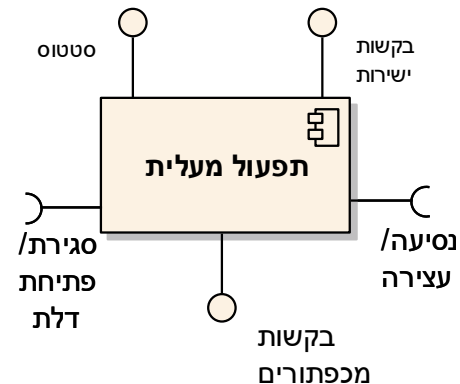
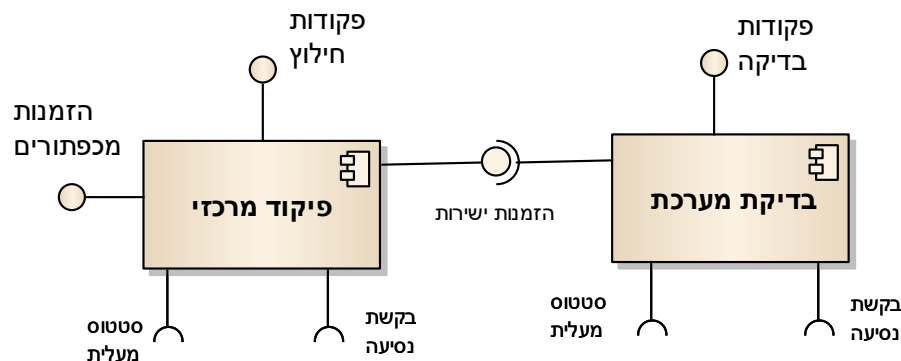
- ישויות חיצוניות שאינן רכיבים אינן מיוצגות בתרשים

- הממשקים החשופים מיועדים לאינטראקציה איתן



טבלת ממשקי תוכנה (SW-ICD)

ממשקים				רכיבים		
פירוט	שירות	סוג	זיהוי	תפקיד	שם	זיהוי
העברת בקשות ע"י כפתורי קומות	הזמנות מכפתורים	מסופק	CCIF-1	בקרה על המערכת הכוללת	פיקוד מרכזי	CC
העברת פקודות מפאנל המחלץ	פקודות חילוץ	מסופק	CCIF-2			
העברת הזמנות שלא דרך כפתורים	הזמנות ישירות	מסופק	CCIF-3			
העברת בקשת נסיעה למעלית	בקשת נסיעה	נדרש	CCIF-4			
בקשת סטטוס מעלית	סטטוס מעלית	נדרש	CCIF-5			
			OEIF-1	בקרה ותפעול של מעלית בודדת	תפעול מעלית	OE
			OEIF-2			
			OEIF-3			
			OEIF-4			
			OEIF-5			
			STIF-1	ביצוע בדיקות למערכת הכוללת	בדיקת מערכת	ST
			STIF-2			
			STIF-3			
			STIF-4			



- בנה component diagram לכל אחד מפריטי התוכנה של המערכת

– חלק כל אחד מפריטי התוכנה לרכיבים פונקציונאליים

- ניתן לסווג את הרכיבים על פי "שכבות פונקציונאליות", כגון:

– שכבת המידע (data layer)

» רכיבים האחראיים לאיסחון ולאיחזור כל הנתונים

– שכבת היישום (application layer)

» רכיבים האחראיים לפונקציות עיבוד/בקרה שונות

– שכבת התצוגה (presentation layer)

» רכיבים של ממשקי-משתמש האחראי לאינטראקציה עם עמדות ההפעלה

– שכבת הבקרה (control layer)

» רכיבים האחראיים לבקרה על התקנים חיצוניים ("דרייברים")

- לכל רכיב הגדר ממשקים מסופקים ונדרשים

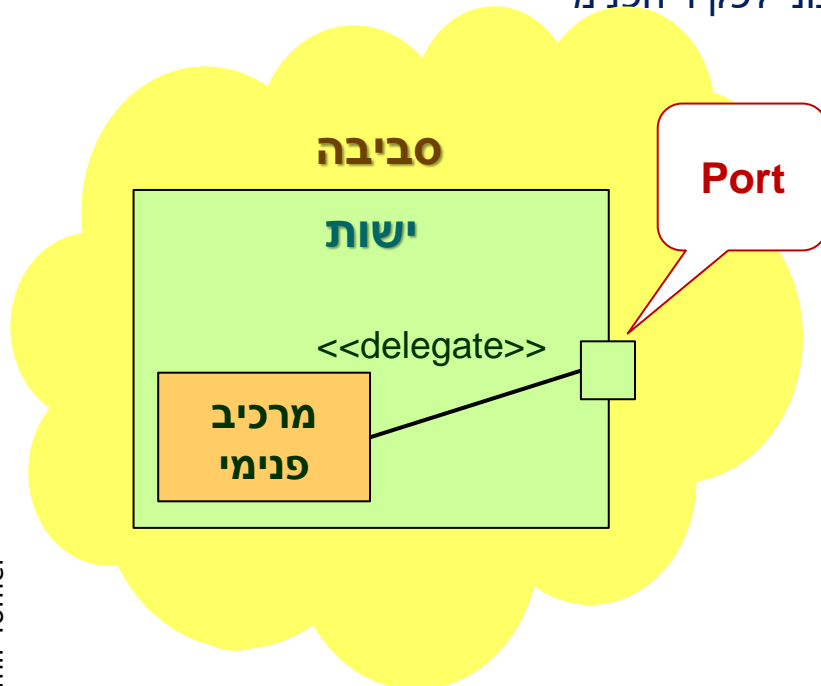
– זהה ממשקים פנימיים (מחברי הרכבה) וממשקים חיצוניים

האצלה (delegation) של ממשקים בישויות מורכבות

- במבנה המערכתי החיבור בין הסביבה או ישות חיצונית לבין ישות או מרכיב פנימי נעשה באמצעות ישות ביניים

– לדוגמה, לקוח נותן הוראה לפקיד הבנק באמצעות מערכת הטלפון

- הטלפון משמש כאמצעי חיבור בין הלקוח החיצוני לפקיד הפנימי



- יציאה (Port)

– נקודת חיבור בין ישות לסביבתה

- דוגמה 1: תקע/שקע או כל מחבר אחר
- דוגמה 2: צומת ברשת תקשורת
- דוגמה 3: חיבור אינטרנט

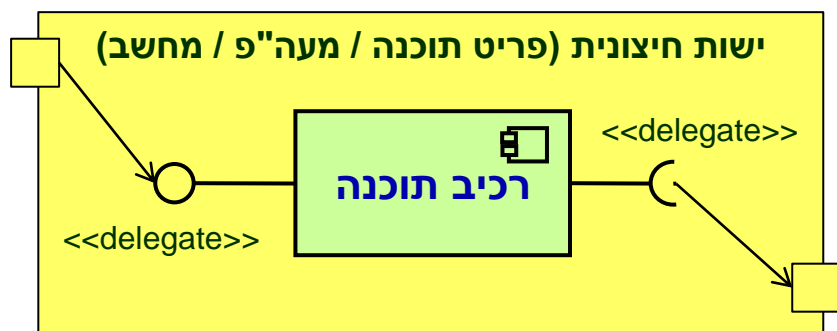
- חיבור האצלה (delegation connector)

– חיבור בין יציאה חיצונית לבין מרכיב פנימי

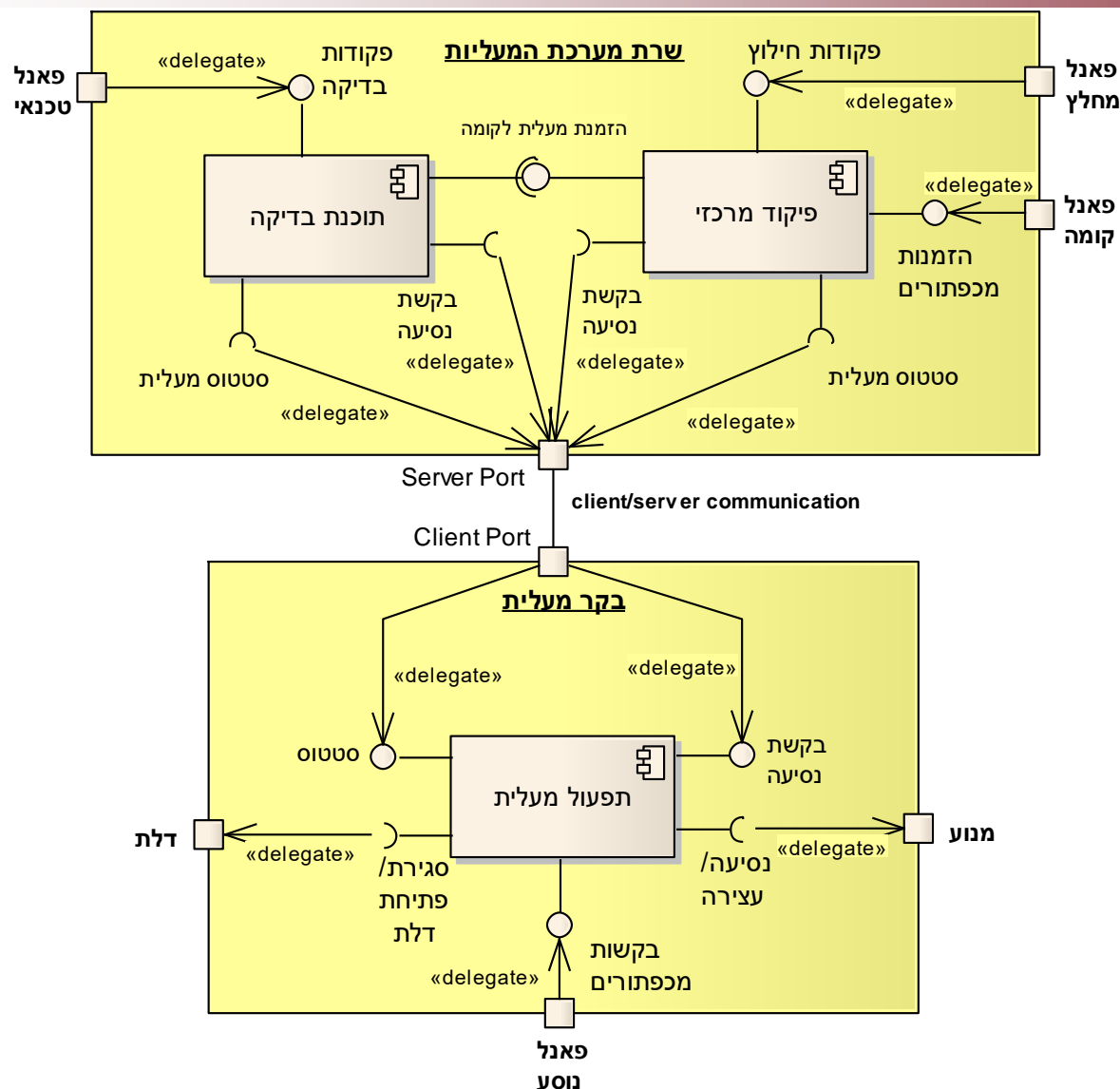
- חיבור יוצא: המרכיב הפנימי משתמש ביציאה לצורך קבלת שרות מהסביבה
- חיבור נכנס: המרכיב הפנימי **מממש** את השרות הניתן דרך היציאה

מימוש ממשקי התוכנה באמצעות חיבורי החומרה

- כאמור, תוכנה תמיד "עטופה" בחומרה
 - פריט תוכנה תמיד מותקן בפריט חומרה (צומת)
 - ממשק בין פריט/רכיב תוכנה לסביבה החיצונית עובר תמיד דרך יציאה/מחבר של חומרה
- רכיבי תוכנה משתמשים ביציאות (ports) לצורך קשר עם סביבתם
 - חיבורי התקנים חיצוניים למחשב (USB, יציאות תקשורת וכו')
 - חיבורים דרך מערכת ההפעלה / התקשורת (קבצים חיצוניים, קלט/פלט, פרוטוקולי תקשורת)
 - חיבורים דרך אמצעים אחרים של הפריט (item) המייצג אותם (דרייברים וכו').
- האצלת ממשק מסופק
 - הרכיב מממש את השרות הניתן לסביבה החיצונית דרך היציאה
- האצלת ממשק נדרש
 - הרכיב מקבל את השרות מהסביבה החיצונית דרך היציאה

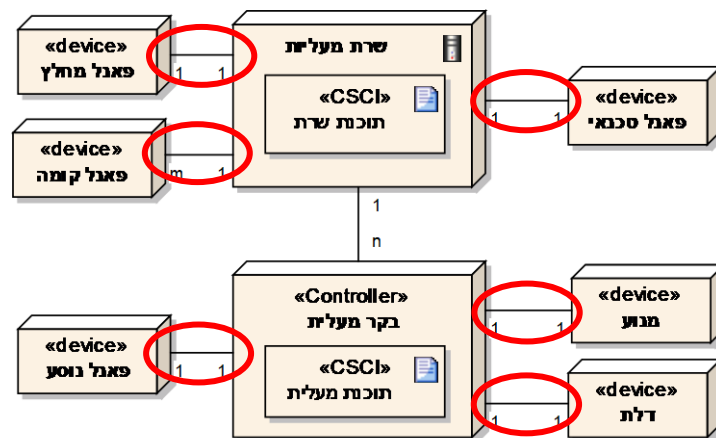
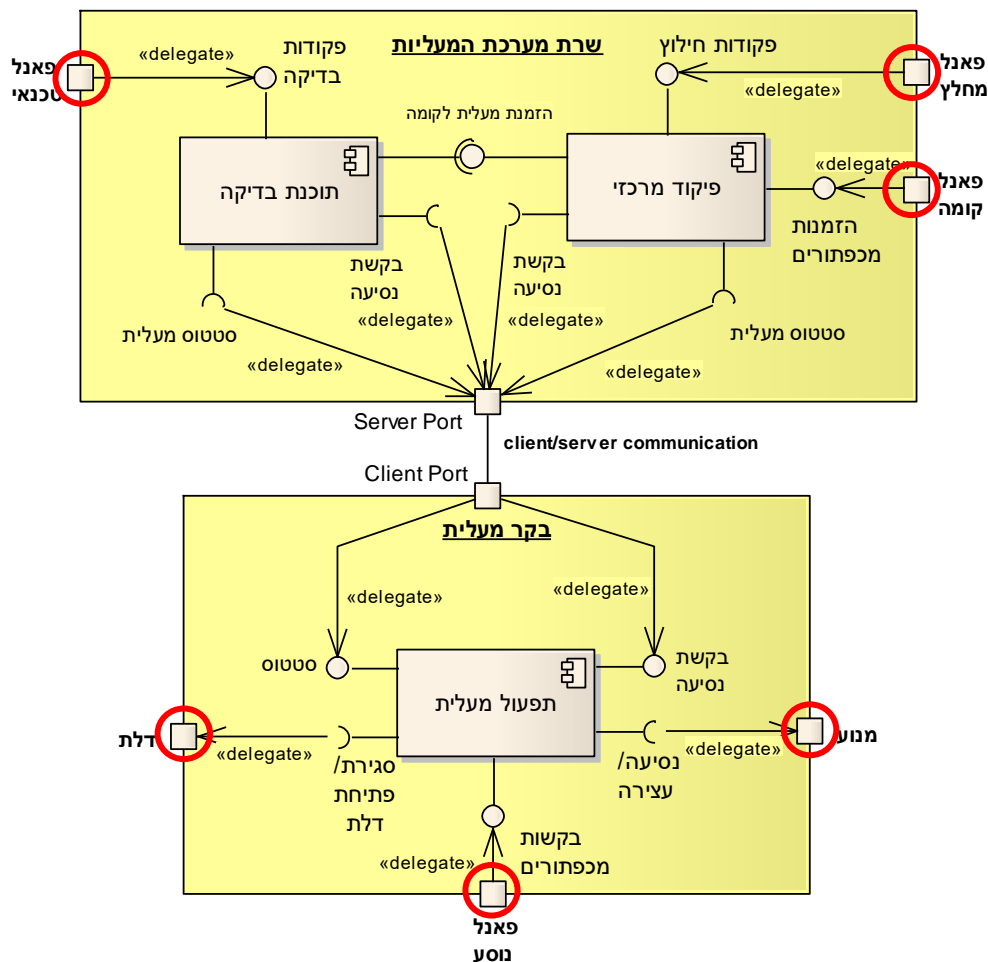


מערכת המעליות – ארכיטקטורת התוכנה בסביבת החומרה



שמירת עקביות (consistency) בין ארכיטקטורת התוכנה לארכיטקטורת החומרה

1. לכל ממשק חיצוני של ארכיטקטורת התוכנה יש port המייצג אותו (delegate)
2. כל port בתרשים ארכיטקטורת התוכנה מזוהה עם קשר פיזי בתרשים הפריסה (deployment diagram) של ארכיטקטורת המערכת

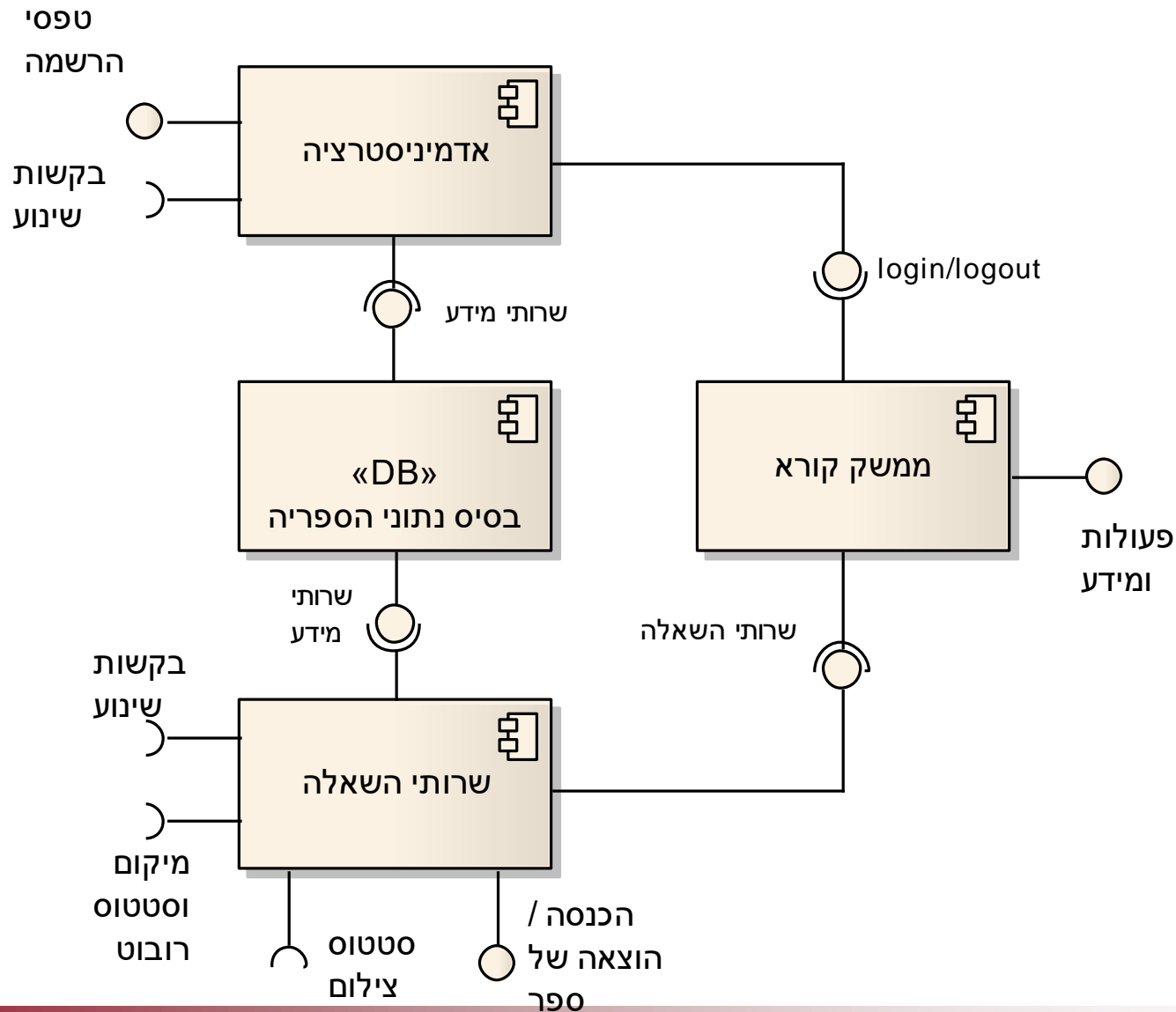


- ערוך תרשים ארכיטקטורה מערכתית של כלל המערכת:

- הגדר את היציאות (ports) של כל צומת
- קשר את היציאות עם הממשקים החיצוניים של ארכיטקטורת התוכנה באמצעות קשרי האצלה (<<delegate>>)

ארכיטקטורת תוכנה

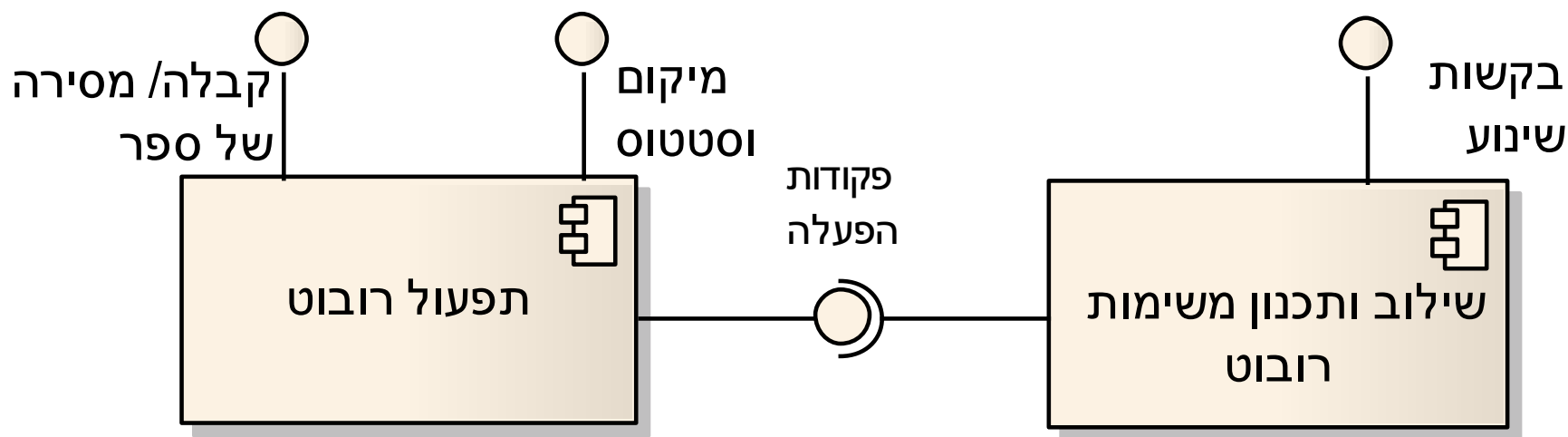
Component Diagram – תוכנת הספרייה



פירוט רכיבים וממשקים

שם	תפקיד	ממשקים		
		סוג	שם	פירוט
אדמיניס-טרציה	פעולות ניהוליות של הספריה	מסופק	טפסי הרשמה	ממשק אינטרנטי לקוראים המעוניינים להירשם כמנויים
		מסופק	login/ logout	הפעלה ועזיבה של עמדה
		נדרש	בקשות שינוע	שינוע ספרים לצורך מיון (החזרה למחסן)
		נדרש	שרותי מידע	אחסון ואיחזור מידע לגבי קוראים וספרים
		מסופק	שרותי מידע	פעולות אחסון ואחזור מידע באמצעות שאילתות
בסיס נתוני הספריה	איחסון כל נתוני הספריה בזמן ומחוץ לשעות הפעילות	מסופק	פעולות ומידע	קלט/פלט מאת/אל הקורא באמצעים השונים (מסך, מקלדת, קורא כרטיסים וכו')
		נדרש	שרותי השאלה	קבלת השרותים השונים של הספריה
		נדרש	login/ logout	הפעלה ועזיבה של עמדה
ממשק קורא	ממשק-המשתמש בעמדת הקריאה	מסופק	שרותי השאלה	קליטת בקשות שרות
		מסופק	הכנסה / הוצאה של ספר	קליטת אירועי הכנסה/הוצאה של ספרים להשאלה מחוץ לספריה
		נדרש	שרותי מידע	אחסון ואיחזור מידע לגבי פעולות השאלה
		נדרש	בקשות שינוע	העברת משימות לרובוט
		נדרש	סטטוס צילום	קבלת מידע לגבי מצב הצילום של ספרים הנמצאים בעמדת הצילום
		מסופק	שרותי השאלה	קליטת בקשות שרות
		מסופק	הכנסה / הוצאה של ספר	קליטת אירועי הכנסה/הוצאה של ספרים להשאלה מחוץ לספריה
שרותי השאלה	הספקת שרותי ההשאלה והצילום לקוראים	מסופק	שרותי השאלה	קליטת בקשות שרות
		מסופק	הכנסה / הוצאה של ספר	קליטת אירועי הכנסה/הוצאה של ספרים להשאלה מחוץ לספריה
		נדרש	שרותי מידע	אחסון ואיחזור מידע לגבי פעולות השאלה

CSCI תוכנת הרובוט – Component Diagram



הערה: הרובוט מסופק כמוצר מדף, אך כאן מובא הפירוט לצורך שלמות ההדגמה

פירוט רכיבים וממשקים

ממשקים			תפקיד	שם
פירוט	שם	סוג	תוכנת ההפעלה של הרובוט לביצוע פעולות נסיעה ופעולות לקיחה והנחה של ספרים	תפעול רובוט
אופני ההפעלה של הרובוט	פקודות הפעלה	מסופק		
זיהוי אירועי קבלה או מסירה של ספר באמצעות קורא בר-קוד	קבלה / מסירה של ספר	מסופק		
קבלת נתונים מהרובוט לגבי מיקומו ומצבו	מיקום וסטטוס	מסופק		
קליטת בקשות למשימות שינוע ספרים ברחבי הספרייה	בקשות שינוע	מסופק	תוכנת תכנון המשימות של הרובוט	שילוב ותכנון משימות רובוט
העברת פקודות הפעלה לרובוט	פקודות הפעלה	נדרש		

RoboLib – ארכיטקטורת תוכנה מערכתית

