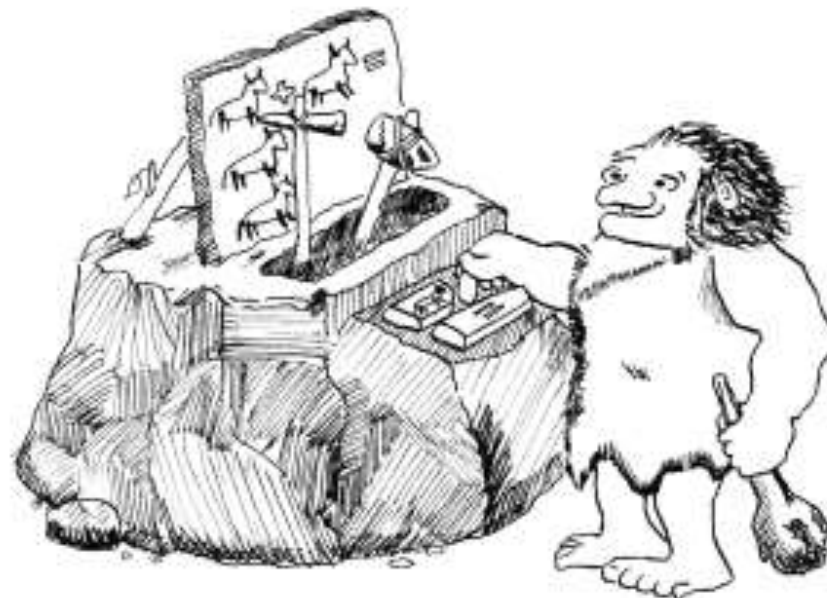


פרק 5

ארכיטקטורה פיזית – הגדרת הפלטפורמה החישובית  
Physical Architecture – the Computational Platform



# פעילות תיכון/תיעוד הארכיטקטורה הפיזית

## • מטרת הפעילות

– תיכון/תיעוד של הארכיטקטורה הפיזית (הפלטפורמה המיחשובית) של המערכת

## • קלט

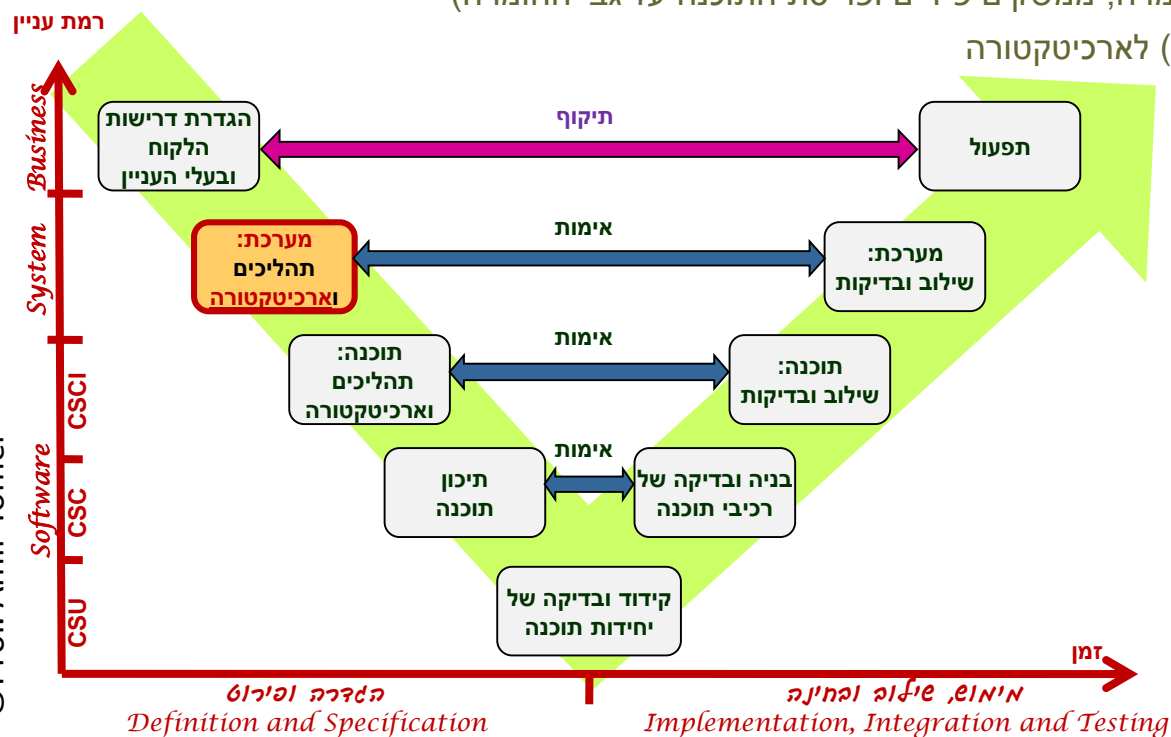
– אילוצי חומרה (HC) מטבלת הדרישות, אפיון טכני של המערכת

– תיכון מערכת/חומרה שנעשה ע"י הנדסת המערכת

## • תוצרים

– מודל הארכיטקטורה הפיזית (חומרה, ממשקים פיזיים ופריסת התוכנה על גבי החומרה)

– עקיבות בין אילוצי החומרה (HC) לארכיטקטורה



# ארכיטקטורה של מערכת עתירת תוכנה

- **מערכת מתאפיינת ע"י:**

- סביבה  
✓ נתונה
- מטרות  
✓ יושגו ע"י תהליכי המערכת שהוגדרו (Use Cases)
- מרכיבים
  - ? מרכיבי חומרה
  - ? מרכיבי תוכנה
- ארגון/מבנה (קשרים פנימיים וחיצוניים)
  - ? קשרים פיזיים
  - ? קשרים לוגיים
- אינטראקציה/התנהגות
  - ? כיצד פועלי הרכיבים במשותף על מנת לממש את התהליכים שהוגדרו

- **לפיכך, ארכיטקטורה של מערכת עתירת תוכנה צריכה לכלול**

- ארכיטקטורה פיזית (מרכיבי חומרה וקשרים פיזיים) – מודל סטטי (מבנה)
- ארכיטקטורה לוגית (מרכיבי תוכנה וקשרים לוגיים) – מודל סטטי (מבנה)
- ארכיטקטורה משולבת (מימוש הקשרים הלוגיים באמצעות הקשרים הפיזיים) – מודל סטטי (מבנה)
- ארכיטקטורת תהליכים (מימוש התהליכים כאינטראקציה בין המרכיבים הלוגיים) – מודל דינאמי (התנהגות)

# ארכיטקטורה פיזית וארכיטקטורה לוגית

- ארכיטקטורה פיזית

- רכיבים פיזיים

- "קופסאות" חומרה

- לדוגמה: מעבדים, התקני איחסון, התקני תקשורת, ...

- ממשקים פיזיים

- חיבורים בין רכיבים פיזיים המאפשרים להעביר באמצעותם מידע

- כבלים, סיבים אופטיים, שידור/קליטה אלקטרומגנטית, ...

- 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, ...



המקרה האופייני  
למערכות משובצות  
רב-תחומיות

- מקרה א': הארכיטקטורה הפיזית קיימת / נקבעה מראש

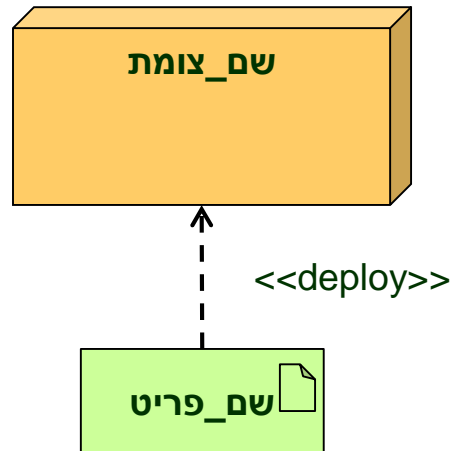
- מזהים רכיבים וממשקים לוגיים
- שוקלים חלופות של הקצאת רכיבים לוגיים לרכיבים הפיזיים בארכיטקטורה הנתונה
- בונים ארכיטקטורה לוגית

- מקרה ב': הארכיטקטורה הפיזית טרם נקבעה

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

המקרה האופייני  
למערכות מידע

# מידול ארכיטקטורה פיזית ב-UML באמצעות תרשים פריסה – Deployment Diagram



סימול אלטרנטיבי



- **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

- הפריטים הם רכיבי השימוש (בזמן ריצה)

- **פריסה (Deployment)**

- כאשר פריט תוכנה מותקן בתוך פריט חומרה
- הוא מקיים מולו תלות הנקראת '<<deploy>>'
- מטעמי נוחות המידול, כאשר ממקמים artifact בתוך node, תלות ה-<<deploy>> מתקיימת באופן אוטומטי

# מידול ארכיטקטורה פיזית באמצעות תרשים פריסה – Deployment Diagram

- 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 ונמצא בשימוש המערכת בזמן הפעלתה

- הפריטים הם רכיבי השימוש (בזמן ריצה)

- פריסה (Deployment)

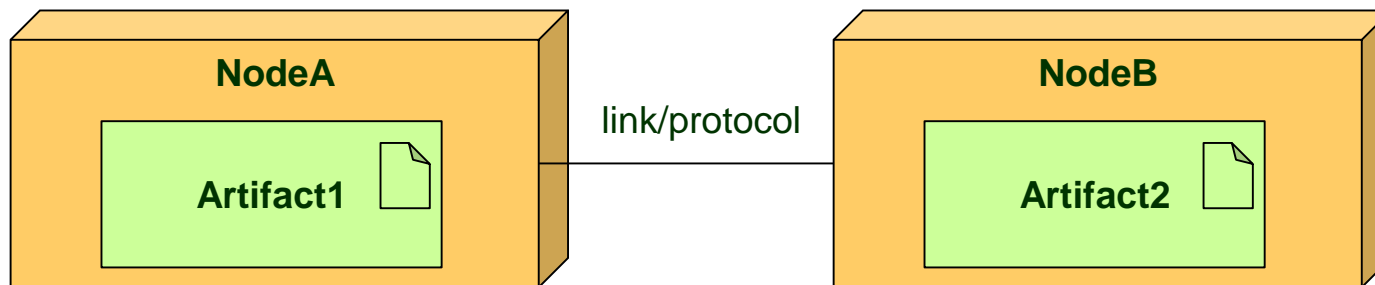
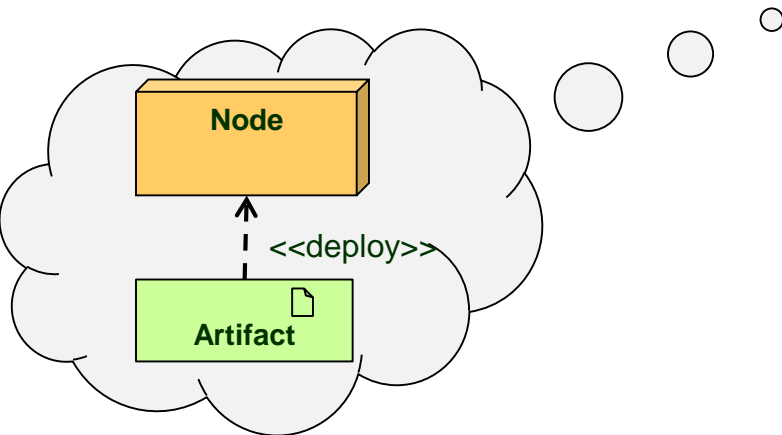
– פריטי התוכנה פרוסים (deployed) בצמתי החומרה

- קשר פיזי

– חיבור (בד"כ דו-כיווני) בין שני צמתים, המתאפיין ע"י

- תווך

- פרוטוקול



## חומרה – צמתים אופייניים

- מחשבים

- שרתים
- מחשבי קצה
- מיקרו-פרוססורים

- התקני איחסון

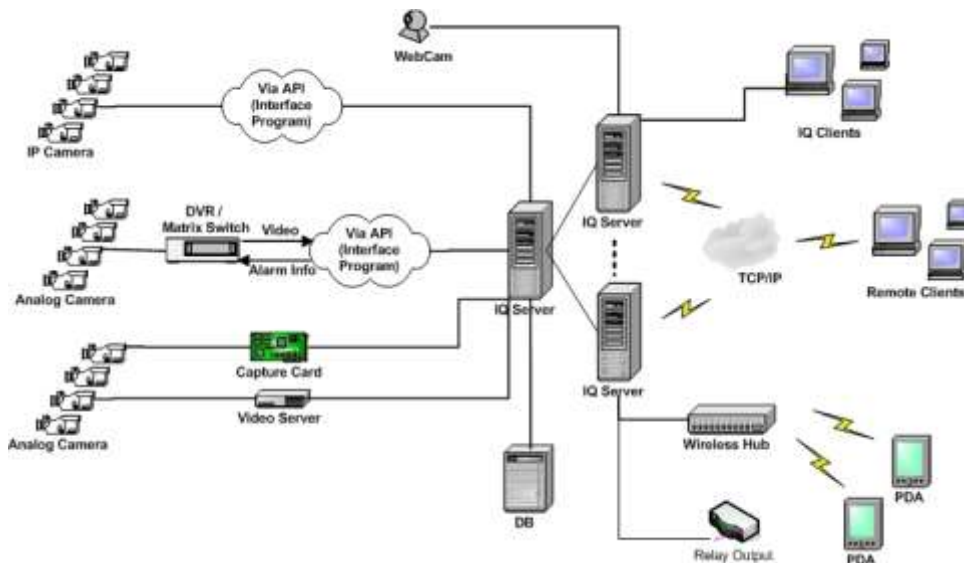
- כוננים
- התקני זכרון חיצוני

- 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**

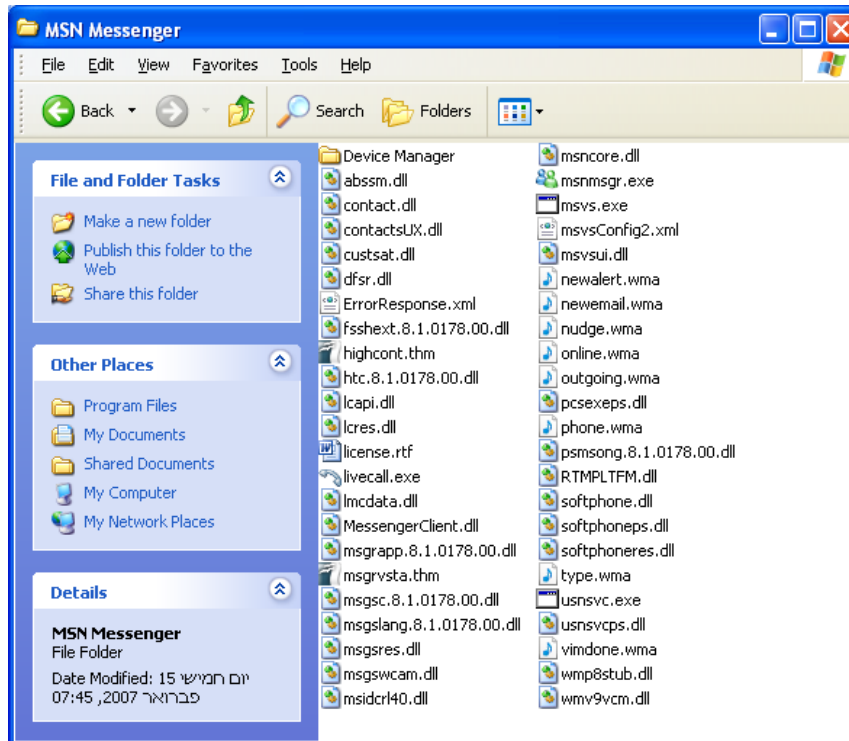
- קבצי נתונים
- בסיסי נתונים

- **פריטי מדיה**

- תמונות
- קבצי אודיו / וידאו

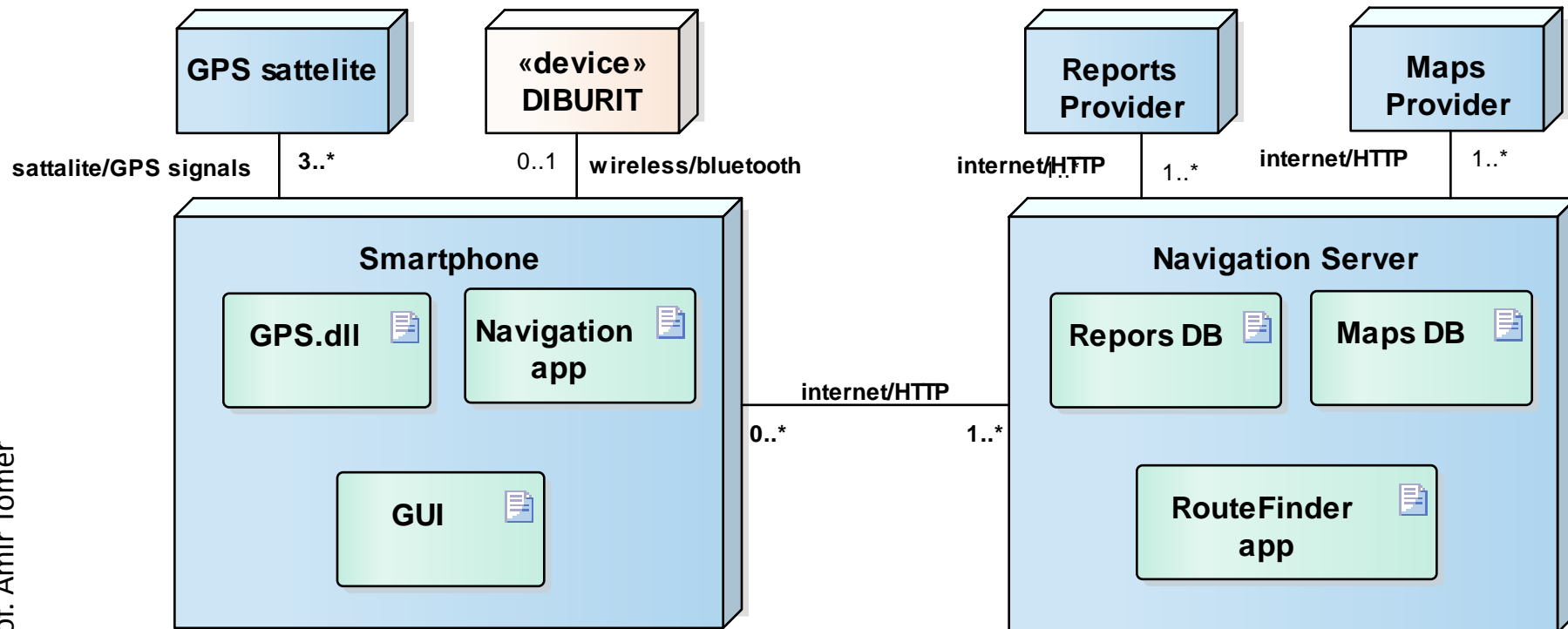
- **פריטי מידע**

- קבצי עזרה
- מדריכים מקוונים



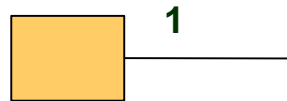
# תרשים פריסה (Deployment Diagram)

- תרשים המתאר את צמתי החומרה, הממשקים שביניהם ופריטי התוכנה הפרוסים בהם.
- לדוגמה: תרשים פריסה (ארכיטקטורה פיזית) של מערכת ניווט

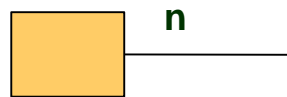


# ריבוי (multiplicity)

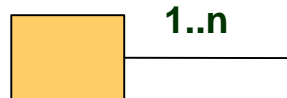
- תחום של שלמים לא שליליים, כולל אינסוף, המגדיר את מספר המופעים של ישות מסוימת.
- הריבוי מופיע בדרך כלל בקצה של קשר, המתאר יחס בין הישות לישות אחרת
- דוגמאות לריבוי:



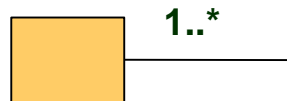
– פעם אחת בדיוק  $[1,1]$



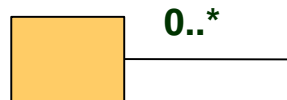
– n פעמים בדיוק  $[n,n]$



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



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

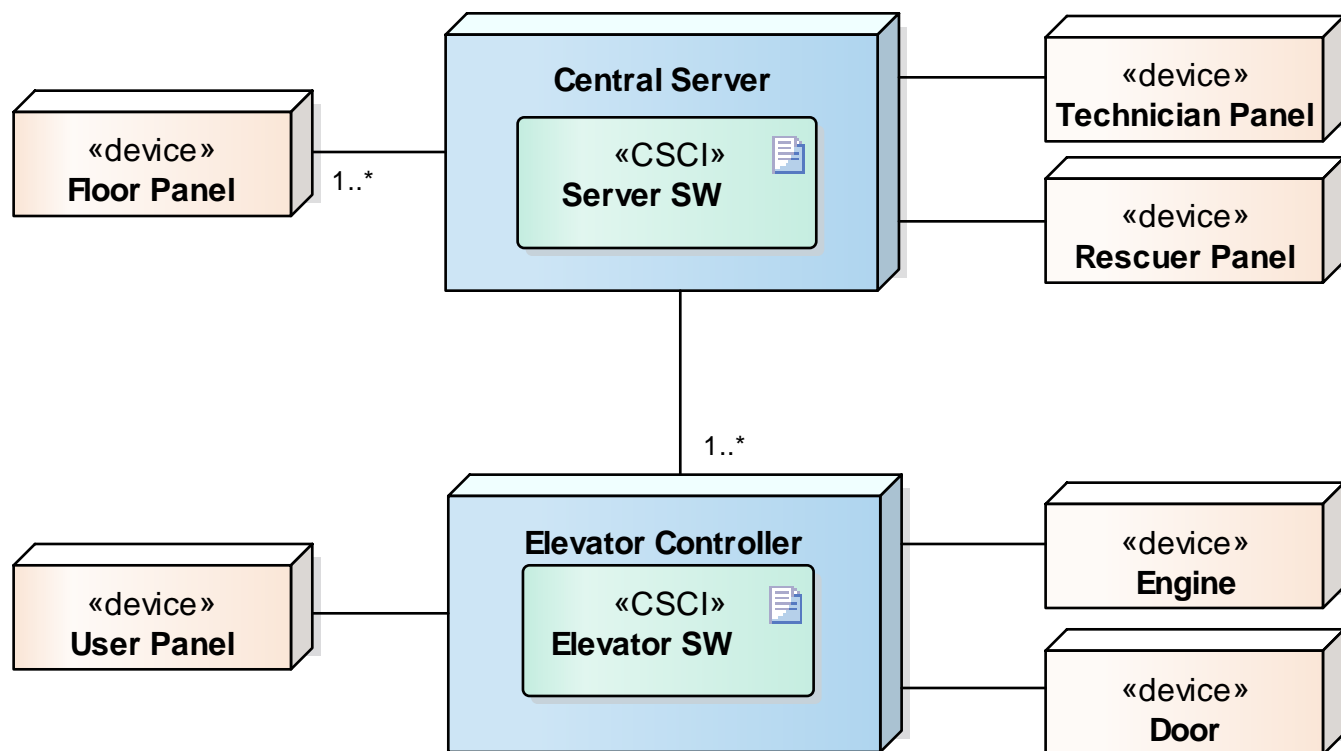


– מספר כלשהו של פעמים (כולל אפס)  $[0..\infty]$

# ארכיטקטורה פיזית א' למערכת המעליות

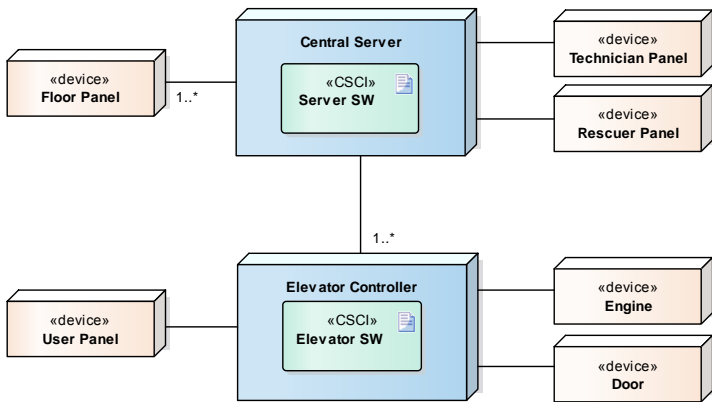
## • ארכיטקטורה מבוצרת

- 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 (ברירת המחדל: כבלים)



# ארכיטקטורה א' – טבלת ממשקים פיזיים

"ICD חומרה"



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

## מטלה: ארכיטקטורה פיזית למערכת ePark

- על בסיס סיפור הלקוח, ותוך התייחסות למפרט הדרישות (בעיקר אילוצי חומרה – HC), הציעו ארכיטקטורה פיזית למערכת ePark באמצעות Deployment Diagram

- בחרו צמתי חומרה והגדירו את הקשרים שביניהם ואת הריבוי המתאים
- נסו לזהות או להגדיר פריטי תוכנה ואת פריסתם על גבי צמתי החומרה

## סיווג דרישות למערכות עתירות תוכנה (2)

### • דרישות לא פונקציונאליות – איכות הפתרון

– דרישות המגדירות תכונות נוספות של הפתרון שצריכות להתמלא תוך כדי מילוי הדרישות הפונקציונאליות (How well?)

### • דרישות ביצועים (PR = Performance Requirements)

– פרמטרים ניתנים למדידה לגבי ביצועי התוכנה

» זמן תגובה, נפח איחסון, ניצולת מעבד וכו'

### • מאפייני איכות (QA = Quality Attributes)

– תכונות המאפיינות את המוצר הכולל

» אמינות (reliability) – פעולה ללא תקלות לאורך זמן

» זמינות (availability) – שירות רצוף, התאוששות מהירה מתקלות

» בטיחות (safety) – הגנה על המשתמשים (והסביבה) מפני המערכת

» אבטחה (security) – הגנה על המערכת מפני משתמשים

» בדיקתיות (testability) – היכולת לבדוק את המערכת ואת פעולותיה (גם בדיעבד)

» אחזקתיות (maintainability) – היכולת לערוך בקלות שינויים ותיקונים במוצר

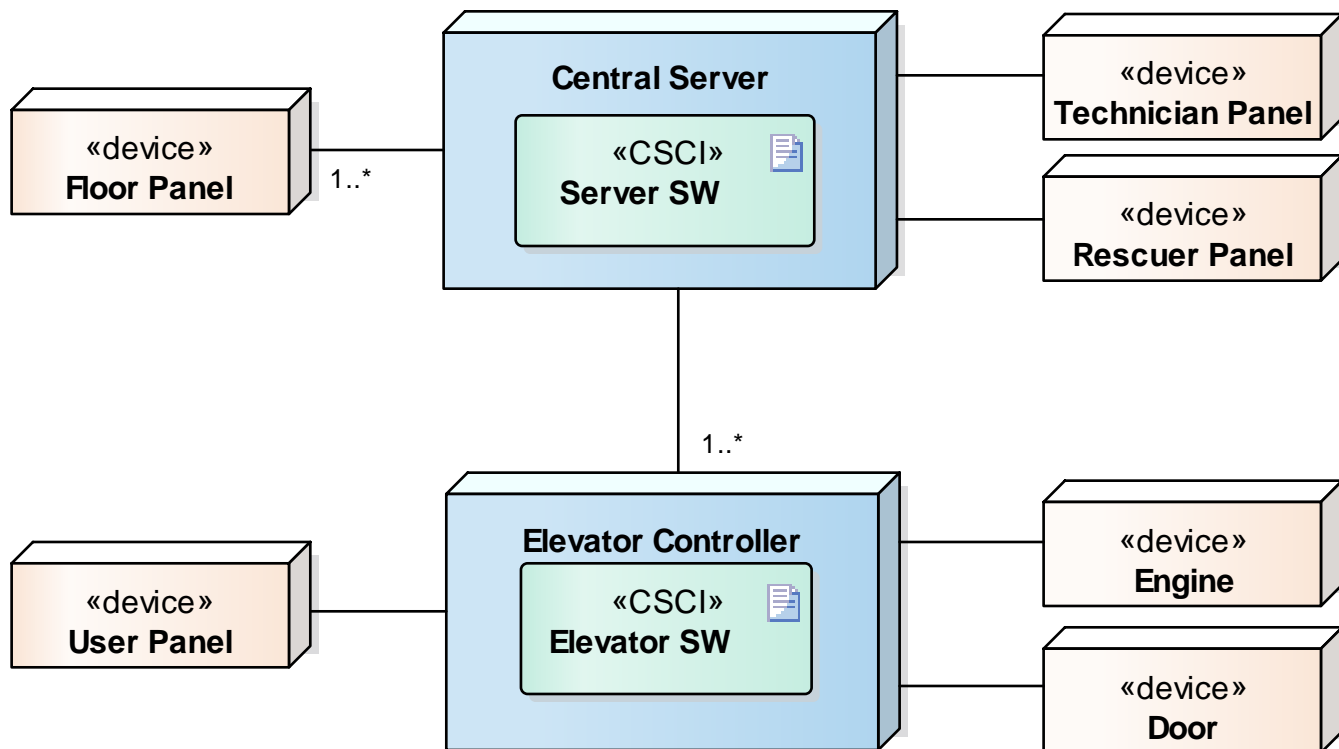
» שימושיות (usability) – האפקטיביות והיעילות שמקנה המערכת למשתמש בביצוע משימותיו ובהשגת מטרותיו

– יש להגדיר את מאפייני האיכות באמצעות מדדים כמותיים!

# ארכיטקטורה פיזית א' למערכת המעליות

## • ארכיטקטורה מבוצרת

- 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

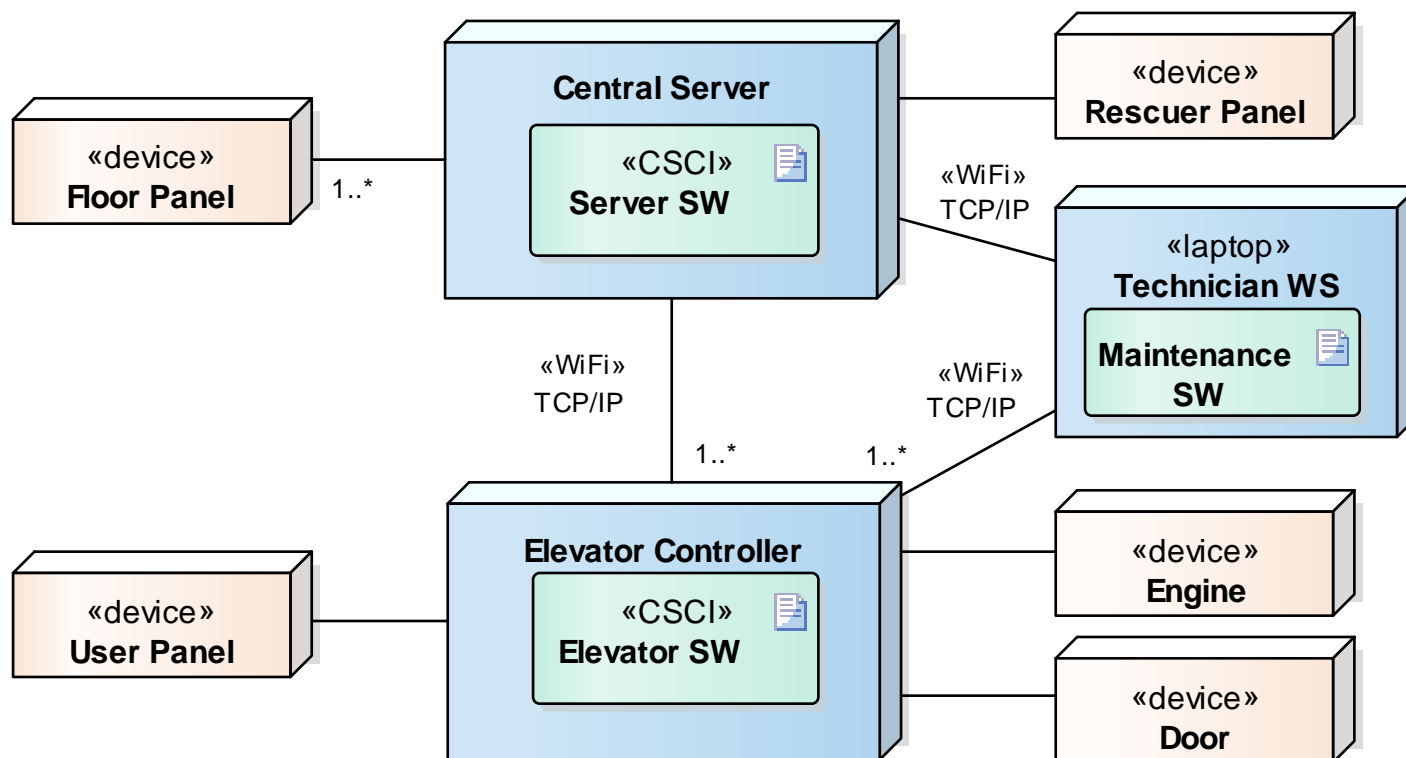




# ארכיטקטורה פיזית ב' למערכת המעליות

## • ארכיטקטורת רשת

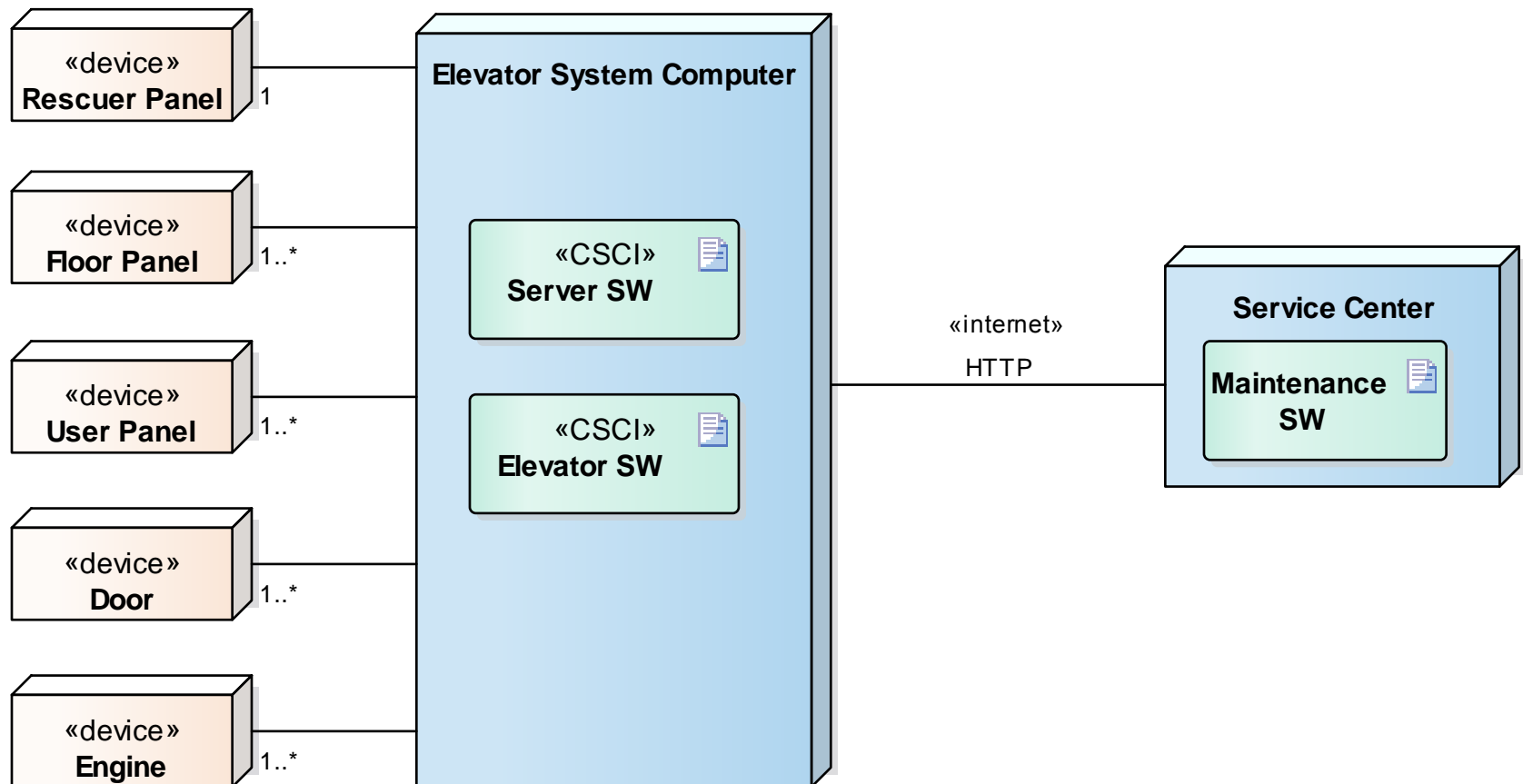
- השרת המרכזי והמעליות מחוברים ברשת תקשורת אלחוטית
- הטכנאי מגיע עם מחשב נייד, המתחבר למערכת דרך הרשת



# ארכיטקטורה פיזית ג' למערכת המעליות

- ארכיטקטורה ריכוזית

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



# הערכת חלופות הארכיטקטורה ע"פ מאפייני איכות

- הארכיטקטורה קובעת את מאפייני האיכות של המערכת
  - הדרישות המשפיעות ביותר על הארכיטקטורה הן הדרישות **הלא-פונקציונאליות**
  - כל הארכיטקטורות מספקות באופן זהה את הדרישות **הפונקציונאליות**

ריכוזית	רשת	מבוזרת	ארכיטקטורה מאפיין איכות
M	M	H	ביצועים
L	M	H	זמינות
H	L	H	אבטחה
H	M	L	תחזוקתיות
L	M	H	עלות