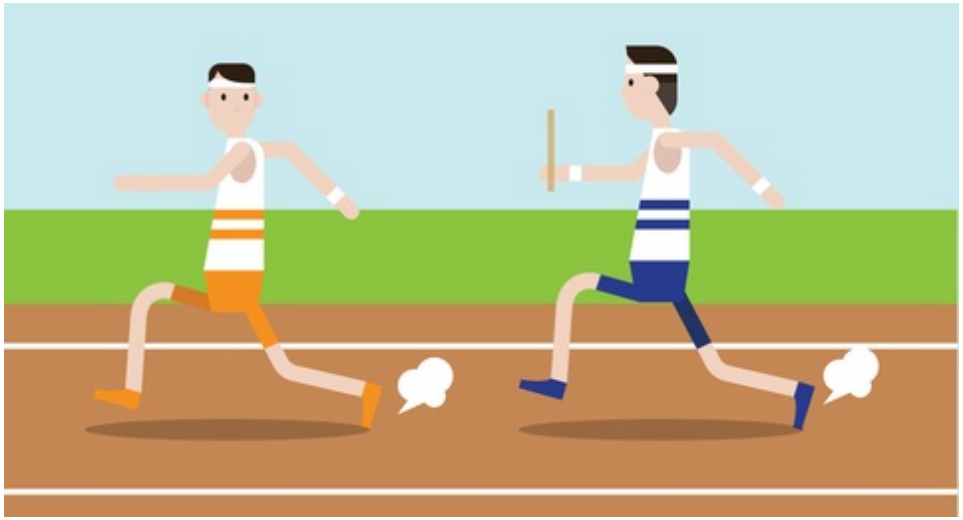
"מירוץ השליחים"

האתגר המשותף של
הנדסת התוכנה והנדסת המערכת

פרופ' עמיר תומר
ראש המחלקה להנדסת תוכנה
– המכללה האקדמית כנרת
הפקולטה למדעי המחשב –
טכניון – מכון טכנולוגי לישראל

amir@amirtomer.com



©Prof. Amir Tomer

המקור העיקרי של ההרצאה

Chapter 5

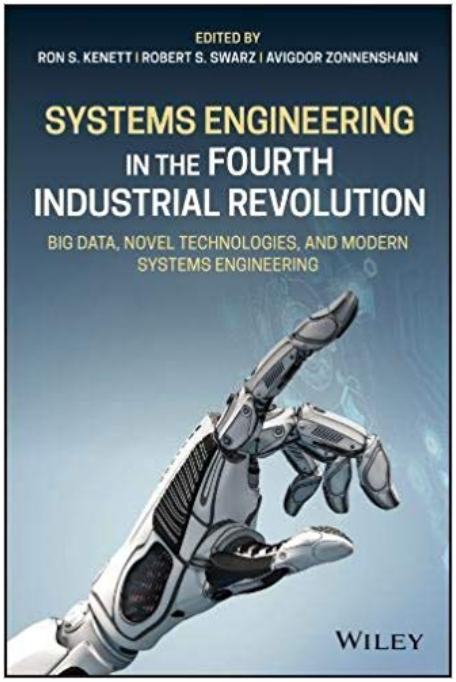
Relay Race: The Shared Challenge of Systems and Software Engineering

Amir Tomer

Synopsis

As systems become more and more software-intensive, the cooperation between systems engineers and software engineers plays a critical role in system development. While many standards (e.g. CMMI, ISO 15288) are aware of this inevitable pairing, they still try to make clear distinction between the two disciplines, with less focus on how to make them work together continuously throughout the system's life cycle. The sensitive points are, as always, the interfaces. There are specific responsibilities that are allocated to each party, but the area of shared responsibilities becomes wider. In that area inter-disciplinary interfaces cannot anymore be implemented as "throwing over the fence" but should rather be perceived as a relay race, where the baton is handed over from one racist to the other through perfect coordination, while both are running very fast. Needless to say that winning the race is a shared challenge, and therefore a baton drop is a shred failure. In this Chapter we analyze the areas of responsibility of the two disciplines and identify those requiring shared effort. Then we propose a detailed process, mainly based upon Functional Analysis, through which a Software-intensive system architecture is constructed, covering both structural and behavioral characteristics.

Keywords: Software-intensive systems, system architecture, software architecture, functional analysis, system life cycle.



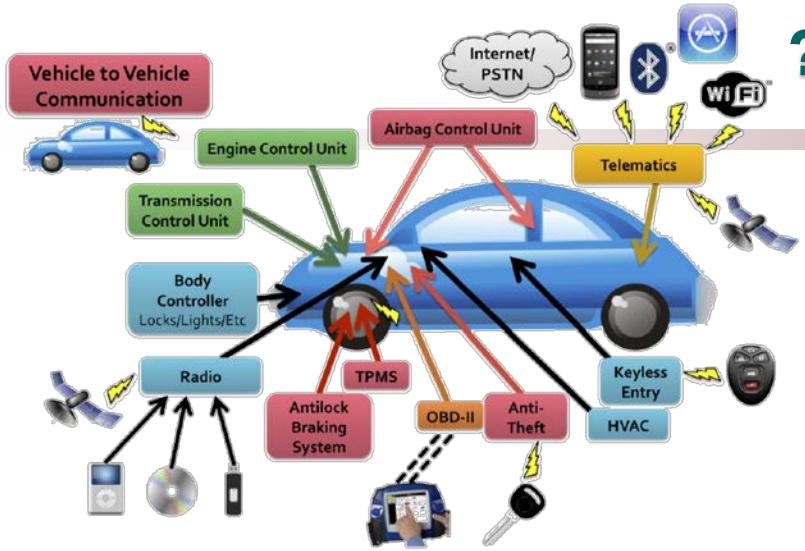
©Prof. Amir Tomer

פרופ' עמיר תומר – הצגה קצרה

- B.Sc. + M.Sc. במדעי המחשב, טכניון
- Ph.D. במיחשוב, Imperial College, London
- 1982-2009: רפאל מערכות לחימה מתקדמות בע"מ
  - מפתח / מוביל / ראש תחום תוכנה
  - מהנדס מערכת שו"ב
  - ממונה על תהליכי הנדסת מערכות ותוכנה בחברה
- 1994-היום: מרצה בטכניון ובמוסדות אקדמיים נוספים
  - הנדסת תוכנה
  - ניהול פרויקטי תוכנה
  - ארכיטקטורת תוכנה
  - הבטחת איכות תוכנה
  - אפיון ותכן מערכות משובצות מחשב (לתואר שני)
- 2009-היום: ראש המחלקה להנדסת תוכנה, המכללה האקדמית כנרת
- הסמכות מקצועיות
  - ניהול פרויקטים: PMP
  - הנדסת מערכות: CSEP
  - הנדסת איכות תוכנה: CSQE
- מאמן מוסמך – אישי/ניהולי

©Prof. Amir Tomer

היכן נמצאת תוכנה במערכת?



http://www.autosec.org/pubs/cars-usenixsec2011.pdf

- ברמת הרכיב/המכלול
  - התוכנה המפעילה רכיב/מכלול מסויים במערכת
    - לדוגמה: ABS
  - בסיס המימוש: תכן תוכנה (software design)
- ברמת המערכת
  - התוכנה המתאמת, מסנכרנת ומאפשרת את פעולתם המשותפת של כל רכיבי המערכת
    - לדוגמה: המחשב המרכזי + רשת התקשורת במכונית
  - בסיס המימוש: ארכיטקטורת מערכת ותוכנה (System and Software Architecture)
- ברמת המערך (System of Systems)
  - התוכנה המאפשרת פעילות משותפת (Interoperability) בין מערכות שפותחו כל אחת בנפרד ולמטרה עצמאית
    - לדוגמה: מכונית+רמזור
  - בסיס המימוש/השימוש: מוסכמות לחילופי מידע (data interchange conventions)

©Prof. Amir Tomer

זווית הראיה של הנדסת המערכת

- הנדסת המערכת עוסקת בכל הכרוך **במכלול השלם**
  - מחזור החיים הכולל
    - הגדרת קונספט, פיתוח, ייצור, תמיכה ותחזוקה
  - המערכת בסביבתה
    - תיאום הממשקים והאינטראקציה של כלל המערכת עם הסביבה החיצונית
  - פירוק (breakdown) והקצאה
    - חלוקת הפונקציונאליות בין חלקי המערכת
    - חלוקת מאפייני האיכות בין חלקי המערכת
  - התאמת המערכת לצרכים ולאילוצים של כלל בעלי העניין (תיקוף – validation)
    - המשתמשים, הלקוח ובעלי העניין האחרים, הארגון המפתח
  - התאמת המערכת לדרישות (אימות – verification)
    - התאמה לדרישות הפונקציונאליות
    - התאמה לדרישות הלא-פונקציונאליות / עמידה במאפייני איכות
  - אינטגרציה כלל-מערכתית
    - הנדסית: תיאום הממשקים והאינטראקציה בין כל חלקי המערכת
    - ניהולית: תיאום הממשקים והאינטראקציה בין קבוצות הפיתוח / הדיסציפלינות השונות

©Prof. Amir Tomer

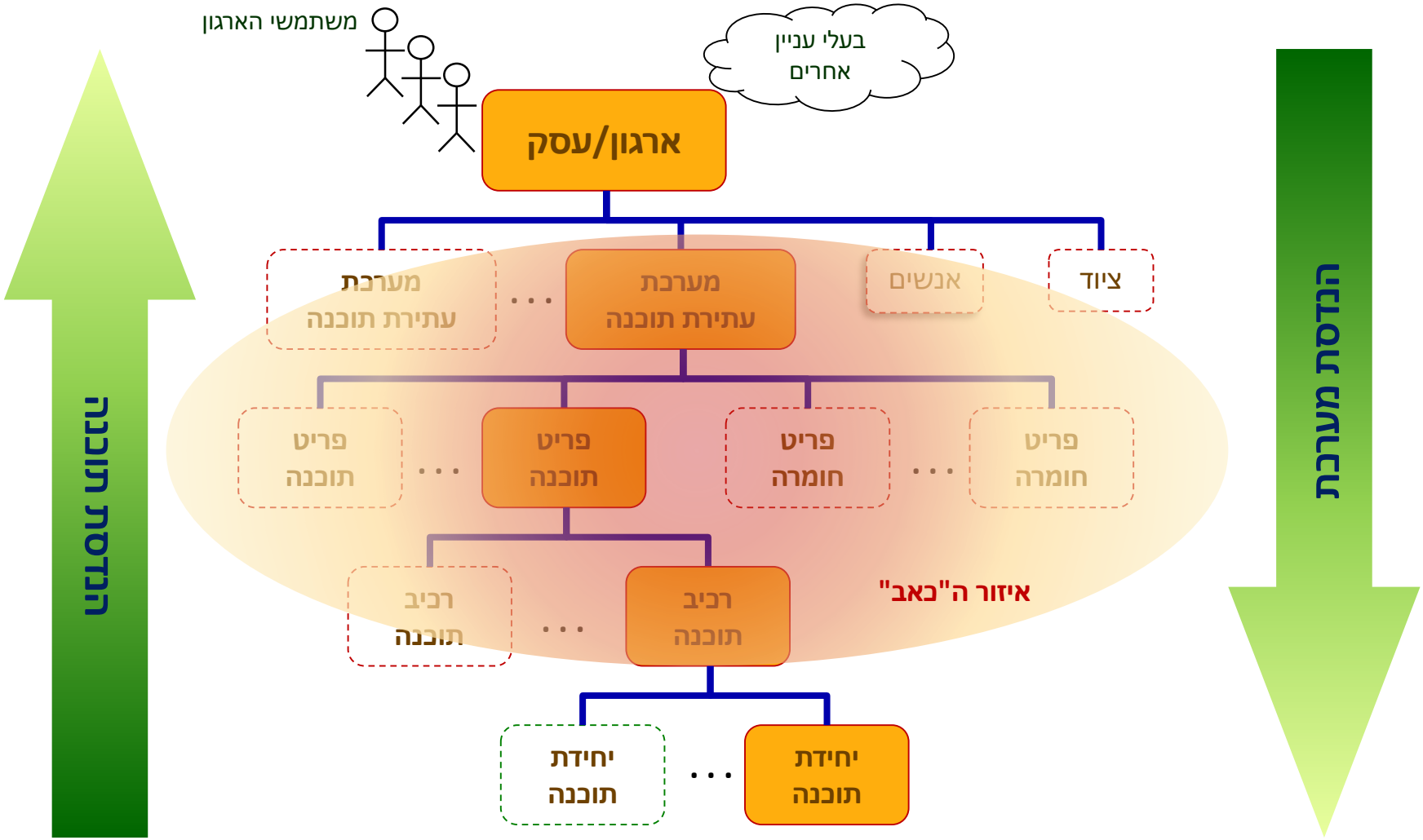
זווית הראיה של הנדסת התוכנה

- לפיתוח תוכנה במערכת עתירת תוכנה יש שני היבטים מערכתיים
  - התוכנה **ב**מערכת
    - מימוש בתוכנה של חלקים של המערכת
    - אינטגרציה של התוכנה עם החומרה
    - פיתוח התוכנה בשילוב עם הדיסציפלינות האחרות
  - התוכנה **כ**מערכת
    - פירוק והקצאה של התוכנה ודרישותיה למרכיבים קטנים יותר
    - פיתוח בין-תחומי באמצעות דיסציפלינות תוכנה שונות
      - GUI
      - DB
      - אלגוריתמים
      - תקשורת
      - ...
    - שילוב ואימות של התוכנה כמערכת כוללת, בסביבת פיתוח תוכנה

©Prof. Amir Tomer

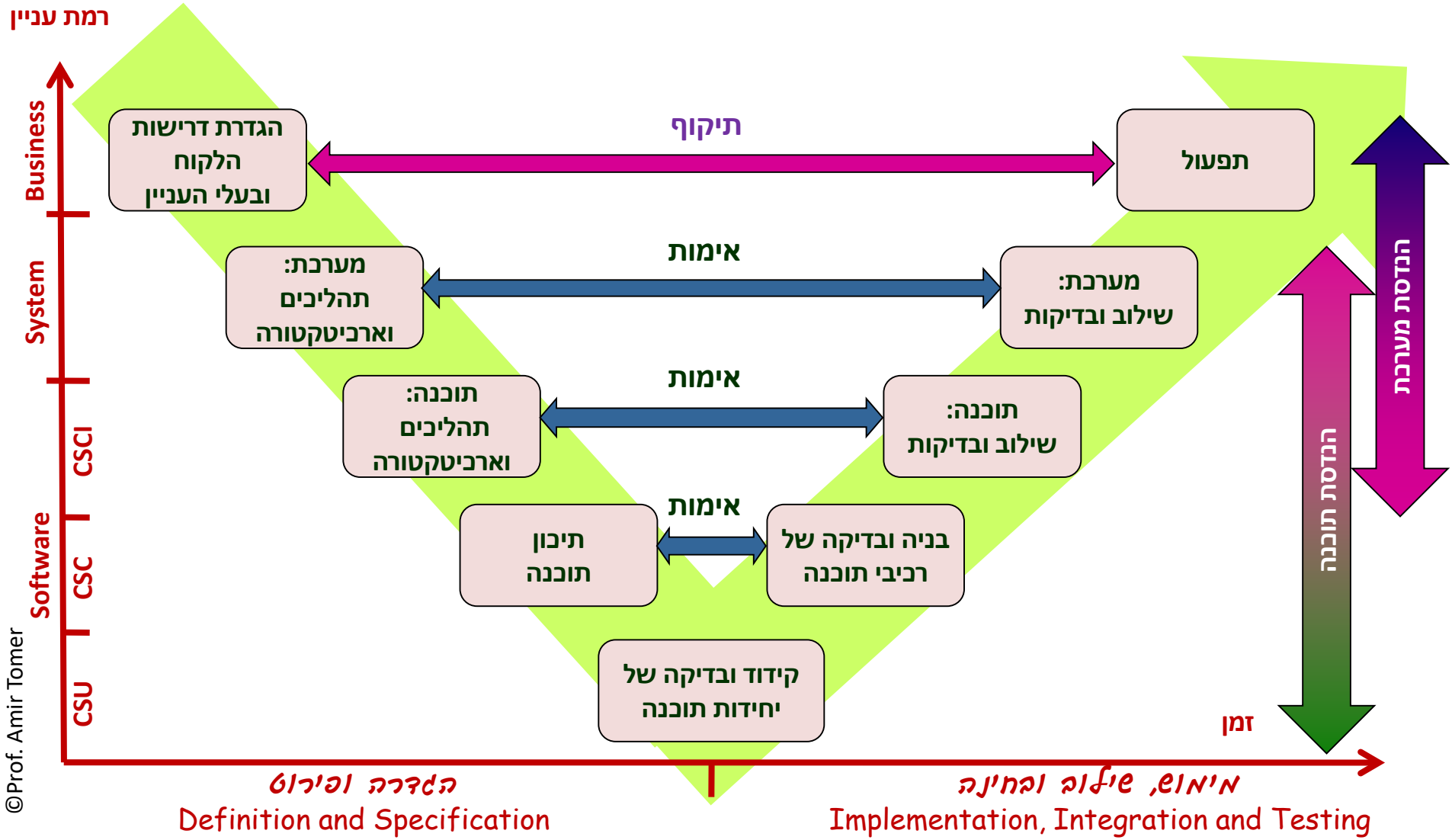
נקודת המבט וחלוקת האחריות המקובלות

"רחוק מן העין – רחוק מן הלב"



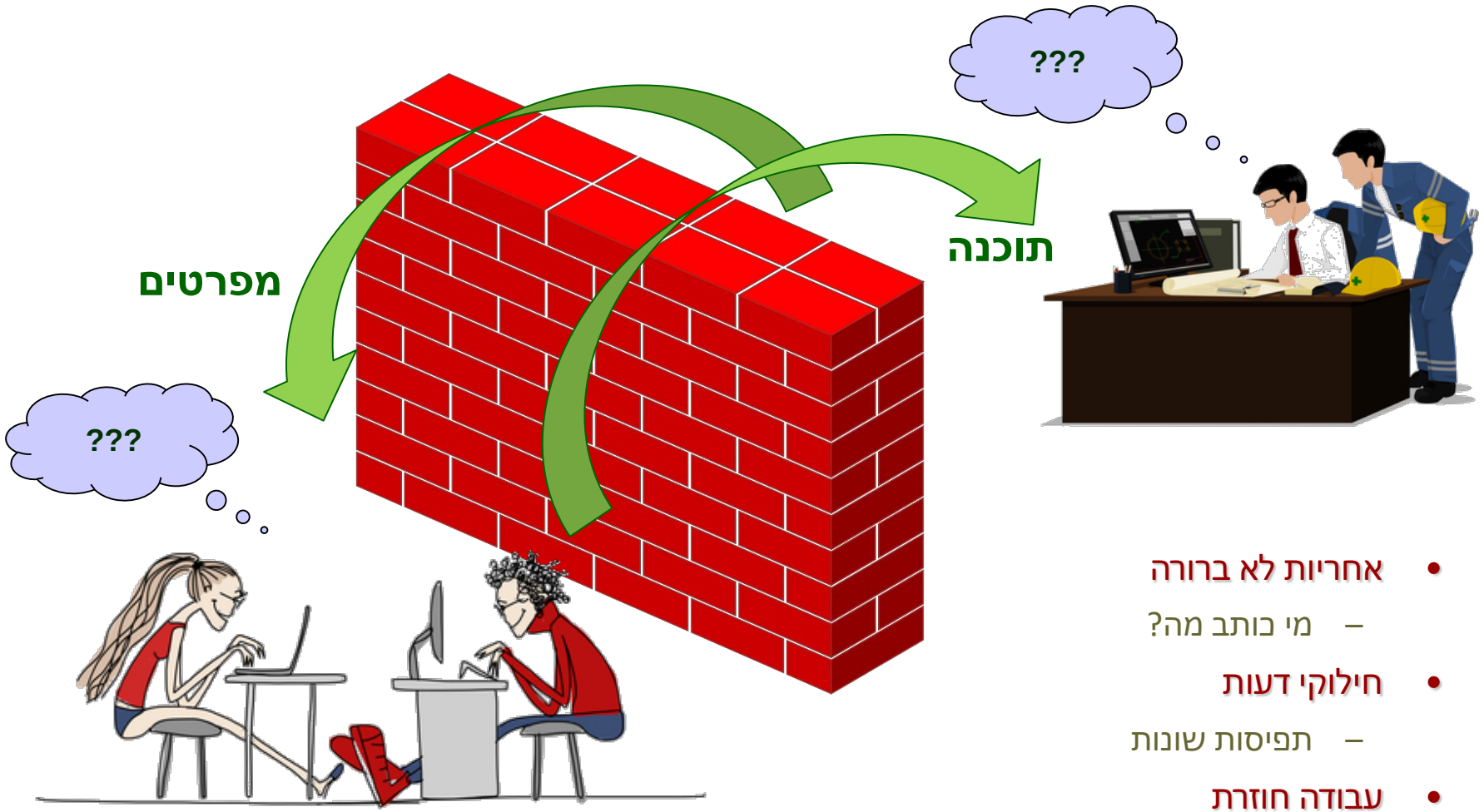
©Prof. Amir Tomer

פיתוח מערכת עתירת תוכנה – פעילויות, תוצרים וחלוקת אחריות



©Prof. Amir Tomer

התוצאות הישירות של חלוקת אחריות שרירותית ("השלכה מעבר לגדר")

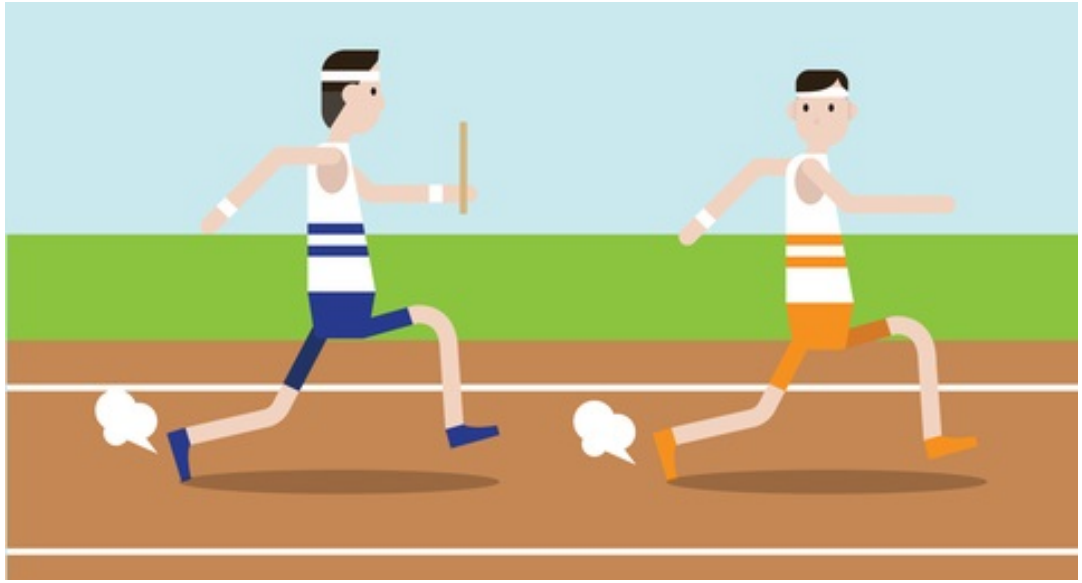


- אחריות לא ברורה
  - מי כותב מה?
- חילוקי דעות
  - תפיסות שונות
- עבודה חוזרת
  - אי-הבנת המפרטים

©Prof. Amir Tomer

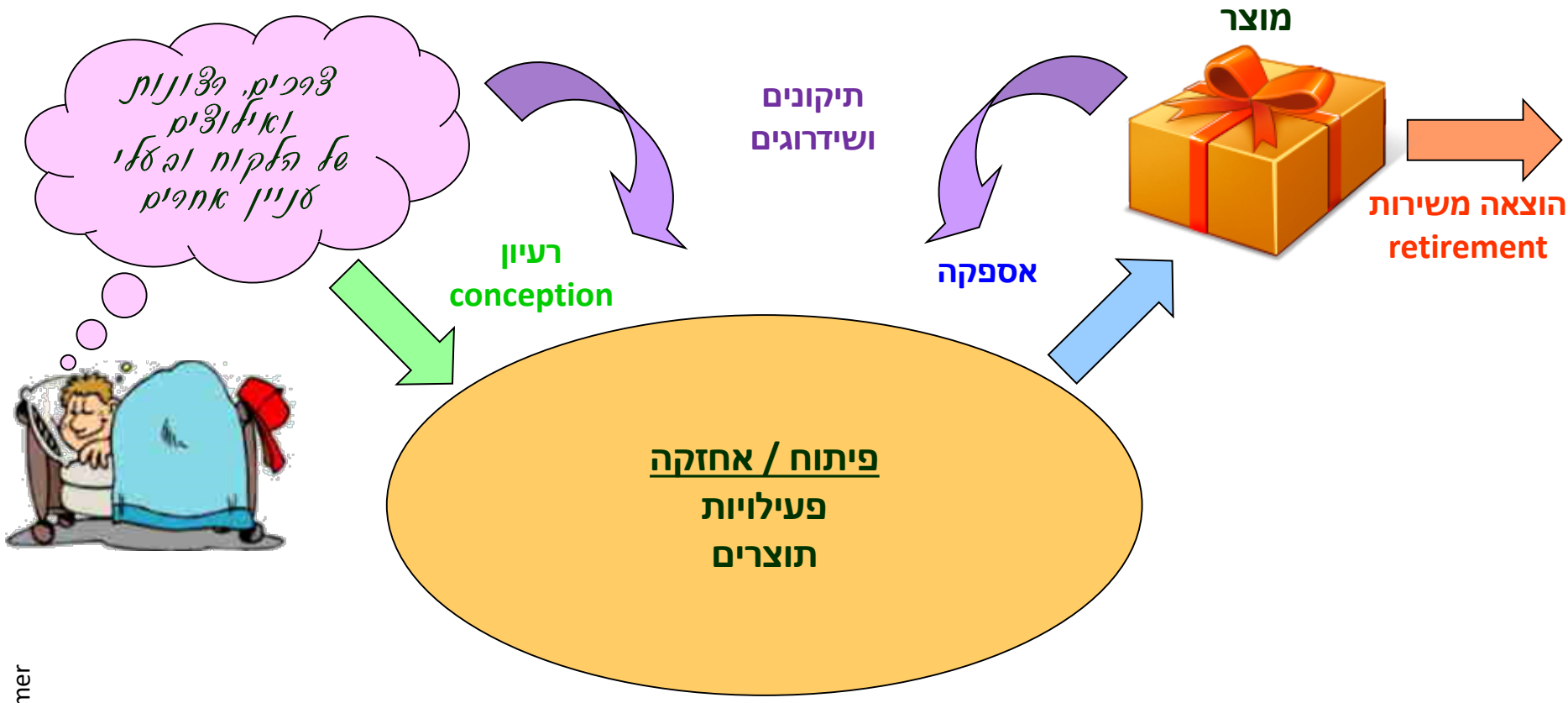
הפתרון הבלתי נמנע – "מרוץ שליחים" (Relay Race)

- הבנה משותפת של המשימה
- אחריות משותפת על אי-שמיטת המקל
- ריצה במקביל
- העברה מובטחת (guaranteed delivery)



©Prof. Amir Tomer

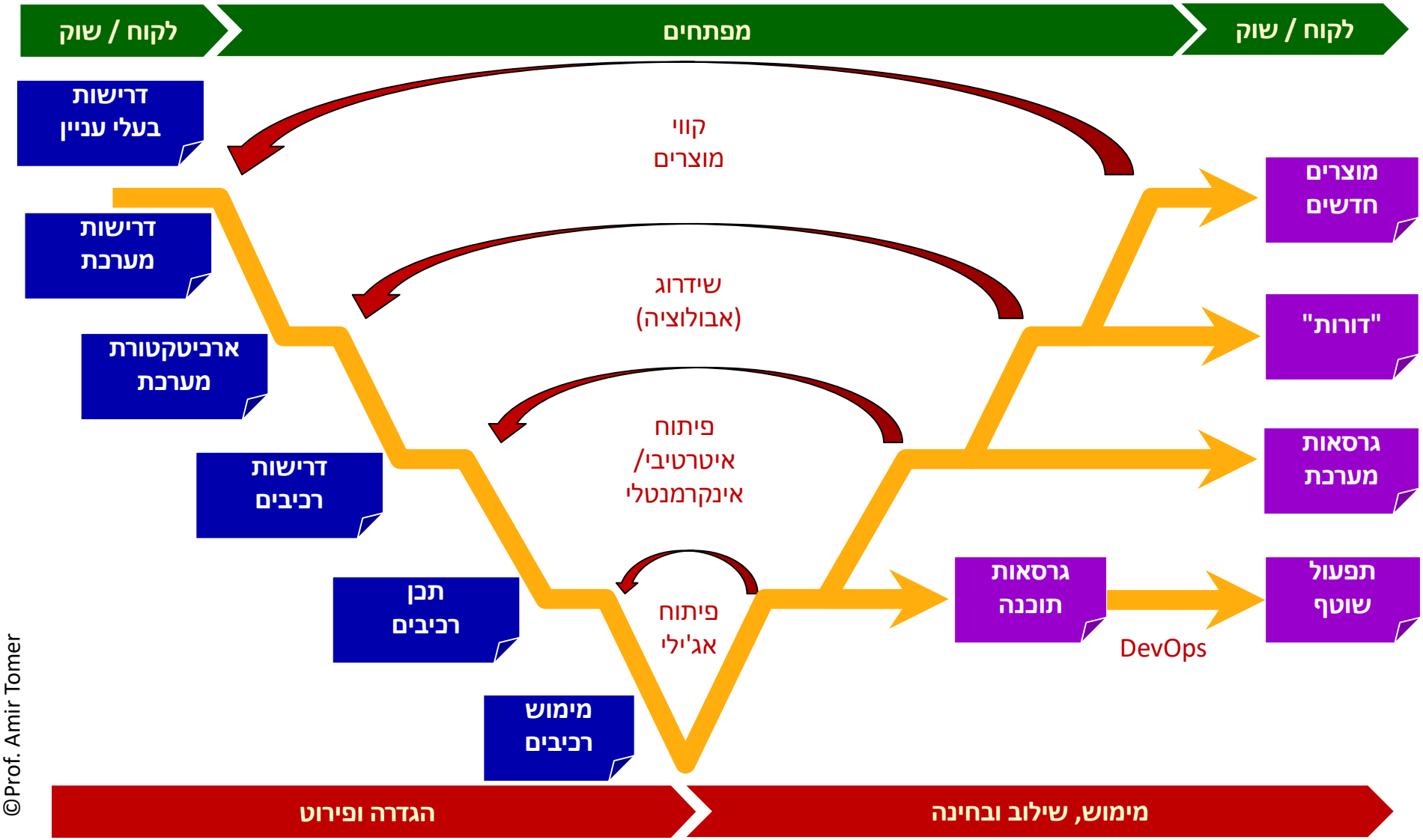
מחזור החיים של מערכת עתירת תוכנה



- מחזור חיים - הגדרה
  - *"אבולוציה של מערכת, מוצר, שרות, פרויקט או ישות אחרת מעשה-ידי-אדם מהרעיון (conception) ועד להוצאה משירות (retirement)"* [ISO/IEC 15288]

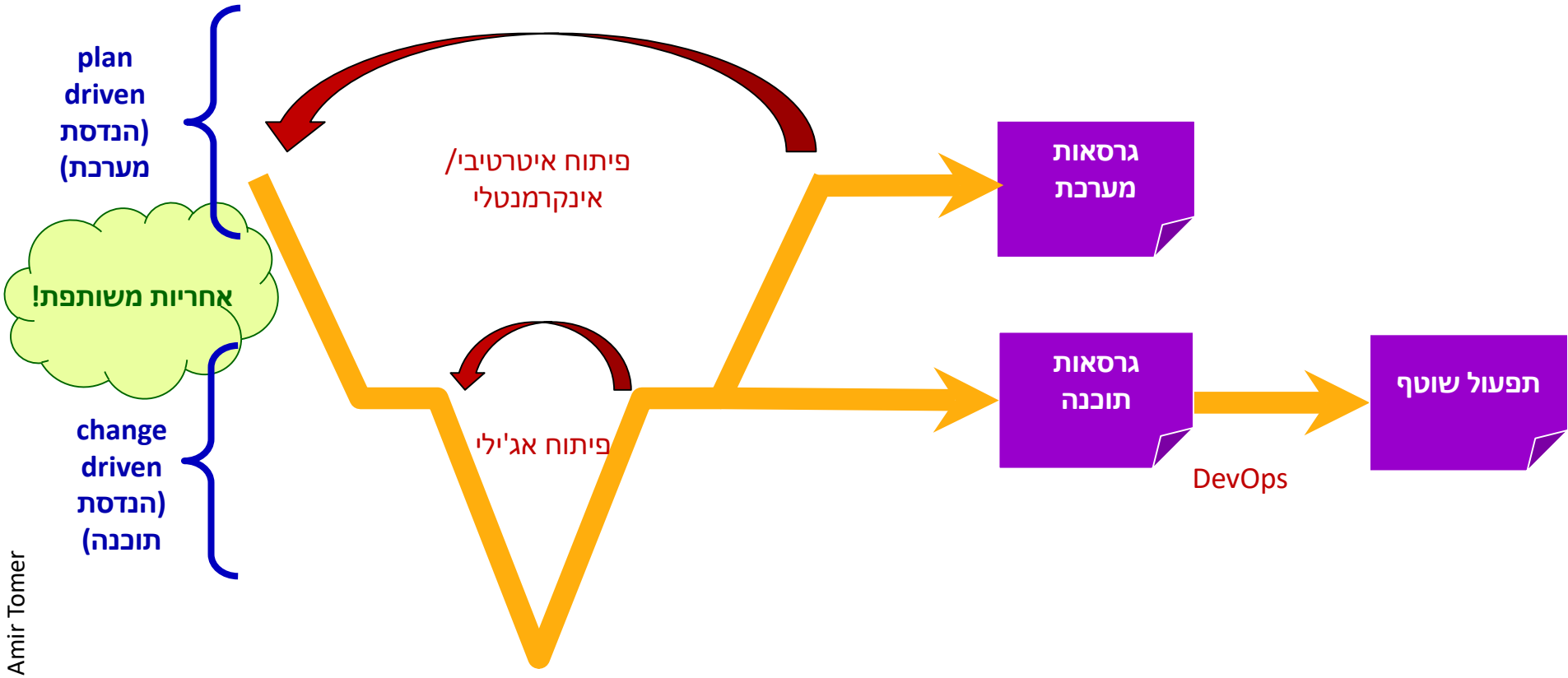
©Prof. Amir Tomer

מחזור החיים האופייני של מערכת עתירת תוכנה – התמונה הכוללת



©Prof. Amir Tomer

חלוקת האחריות על מחזור החיים בפיתוח מערכת בודדת



©Prof. Amir Tomer

מהי ארכיטקטורה מערכתית?

- A **system** is a combination of **interacting elements organized** to achieve one or more stated **purposes**

  [ISO/IEC/IEEE 15288:2015 Systems and software engineering--System life cycle processes, 4.1.46]

- בלומר, מערכת מוגדרת על פי ארבעה מאפיינים:
  - **האלמנטים** (elements) המרכיבים אותה
  - האופן בו הם מסודרים/**מאורגנים**
    - המבנה המערכתי
  - **האינטראקציה** שהם מקיימים ביניהם ועם סביבתם
    - ההתנהגות המערכתית
  - **מטרות** המערכת (purposes) שעבורן הם פועלים

©Prof. Amir Tomer

מה כוללת הארכיטקטורה המערכתית?

- במערכת דינאמית (עתירת תוכנה) יוגדרו מאפייני הארכיטקטורה בדלהלן:
  - המטרות
    - תהליכי המערכת, תפוקותיהם ותועלותיהם
  - האלמנטים
    - מרכיבי חומרה
    - מרכיבי תוכנה
  - הארגון/המבנה
    - ממשקים פיזיים (פנימיים וחיצוניים)
    - ממשקים לוגיים (פנימיים וחיצוניים)
    - פריסה (deployment)
  - האינטראקציה
    - האופן בו האלמנטים פועלים במשותף במהלך ביצוע התהליכים, תוך כדי
      - אינטראקציה חיצונית – עם הסביבה
      - אינטראקציה פנימית – בין המרכיבים

©Prof. Amir Tomer

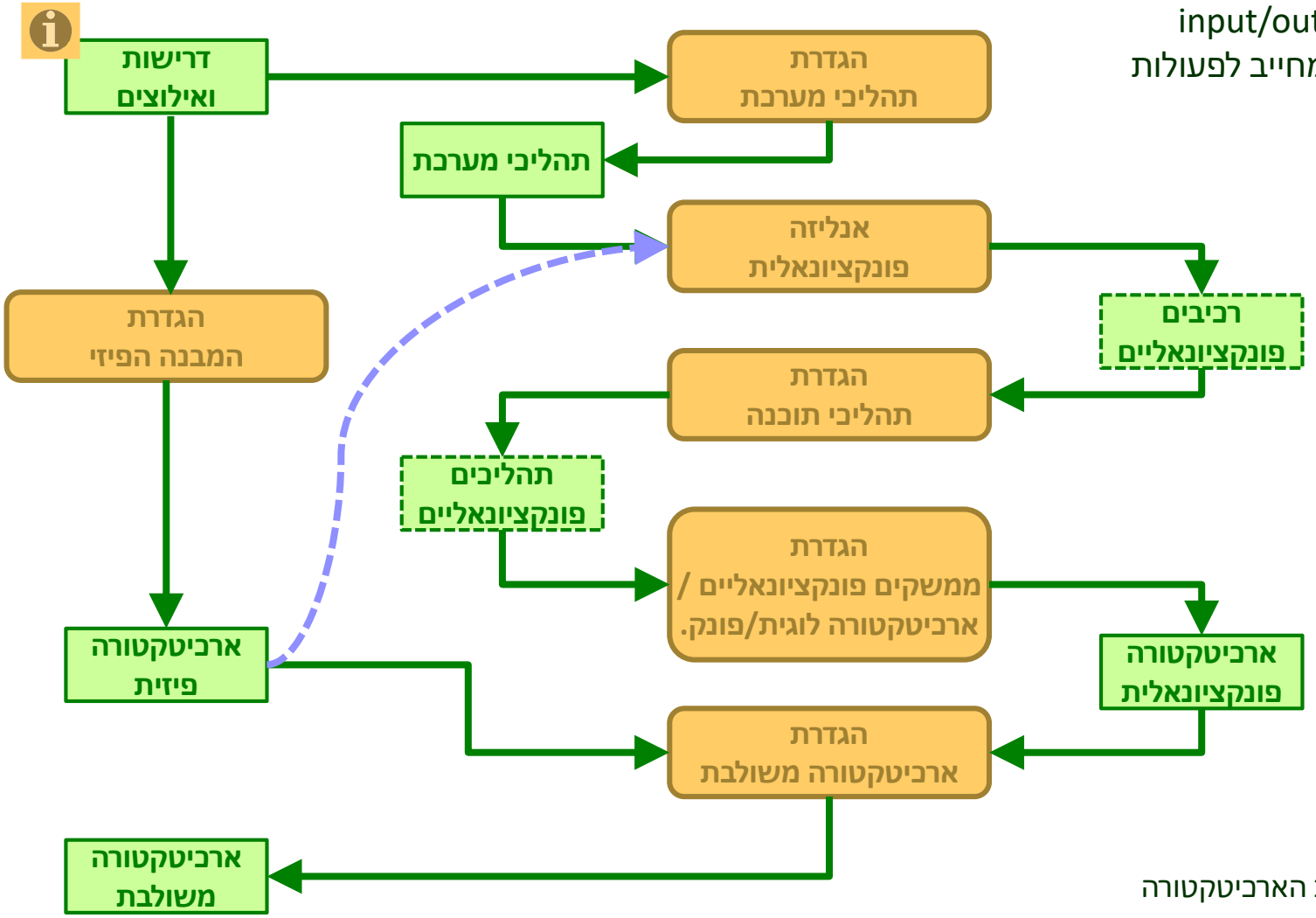
כמה עובדות לגבי ארכיטקטורה...

- הדרישות המשפיעות על הארכיטקטורה הן בעיקר הדרישות **הלא-פונקציונאליות / מאפייני האיכות**
- הפונקציונאליות הדינאמית של המערכת (תהליכי המערכת) מושגת בעיקר ע"י התוכנה
  - אבל התוכנה תמיד מותקנת על חומרה
- ארכיטקטורה היא "הפונקציה היוצרת" של כל התצורות (configurations) המערכתיות
  - תצורה מבצעית, בתרחישים שונים
  - תצורת בדיקות
  - תצורת סימולציה
  - ...
- **מבנה המערכת** נגזר בדרך כלל **מההתנהגות הנדרשת ממנה**, ולא להיפך

©Prof. Amir Tomer

תרשים זרימה* לבניית ארכיטקטורה מערכתית

* זרימת מידע = input/output
פרט לכך אין סדר מחייב לפעולות



©Prof. Amir Tomer

Case Study מבוסס מודלים של מערכת מעליות

- בשקפים הבאים תוצג הדגמה של התהליך, הכוללת
  - פירוט של הפעולות והתוצרים
  - התרומות של הנדסת המערכת והנדסת התוכנה לכל שלב בתהליך
  - מודל בסיסי (UML/SysML) של התוצרים

©Prof. Amir Tomer

הגדרת תהליכי המערכת

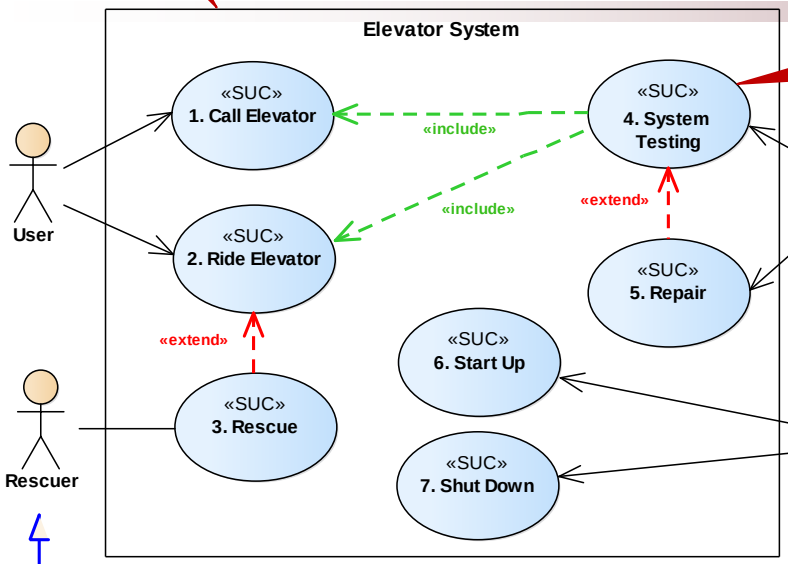
- מהות הפעילות
  - ניתוח דרישות המערכת ע"י תיאור התהליכים המערכתיים, הכוללים:
    - לוגיקת התהליכים, תוך אינטראקציה בין המערכת לסביבתה החיצונית
      - סביבה חיצונית נעזרת: מי/מה שמפעיל את המערכת לצרכיו
      - סביבה חיצונית תומכת: מי/מה שמופעל ע"י המערכת לצרכיה
    - תפוקות ותועלות לבעלי עניין נוספים
    - זיהוי סתירות/חסרים, זיהוי מאפייני איכות*
    - עקיבות (traceability) לדרישות המערכתיות
- תרומת הנדסת המערכת
  - ראיית מערכת כוללת, המערכת בסביבתה, תועלות עסקיות
  - מאפייני איכות מערכתיים
- תרומת הנדסת התוכנה
  - דינאמיקה של תהליכים
  - השלכות מאפייני האיכות המערכתיים על המימוש בתוכנה
  - השלכות השימוש בטכנולוגיות IT מתקדמות (בסיס נתונים, IoT, למידת מכונה, ...)

** Tomer, A. Functional Angels and Quality Devils: Incorporating Quality Scenarios into Functional Scenarios for Software-intensive System Architecture, **Int J Comput Softw Eng** 2019, 4*

©Prof. Amir Tomer

Use Case Model: מודל להגדרת תהליכי המערכת

גבולות המערכת



תהליכי מערכת - הגדרה

סביבה נעזרת / תומכת ("שחקנים")

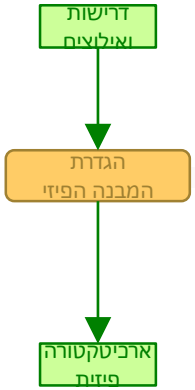
Use-case Diagram

תהליכי מערכת
- פירוט תרחישים
- אינטראקציה

Use-case Specification

SUC-1	הזמנת מעלית
שחקנים ויעדים	נוסע: לקבל מעלית זמינה לנסיעה
ב"ע ואינטרסים	אין
Pre-conditions	• הנוסע נמצא בקומה כלשהי בה נמצאת דלת של מעלית • המערכת פעילה [.Post-cond של UC "איתחול מערכת"]
Post-Conditions	• מעלית פתוחה נמצאת בקומה בה נמצא הנוסע (יעד)
Trigger	• **הנוסע** לוחץ על כפתור עליה / ירידה בקומה
MSS	1. **המערכת** קולטת את הלחיצה 2. **הכפתור** נדלק 3. **המערכת** מאתרת מעלית הנוסעת בכיוון המבוקש 4. **המערכת** מקצה את העצירה למעלית 5. **המעלית** מגיעה לקומה 6. **דלת** המעלית נפתחת 7. **כפתור** הקומה כבה
הסתעפות א'	חלופה בצעד 2 של MSS: הכפתור כבר דלוק (כבר הוקצתה מעלית) 2א1. מעבר לצעד 5
עקיבות לדרישות	...

©Prof. Amir Tomer

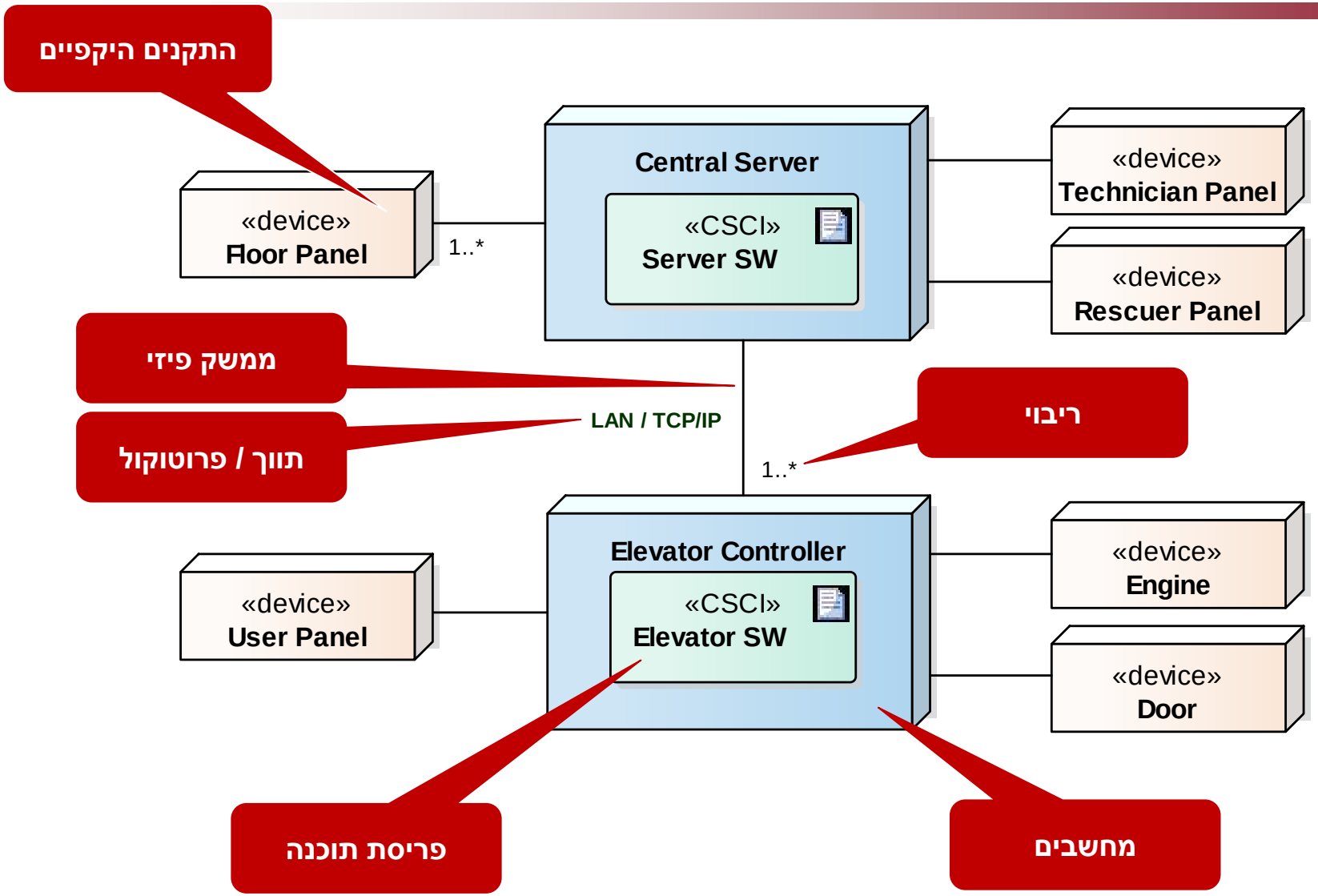


הגדרת הארכיטקטורה הפיזית

- מהות הפעילות
  - הגדרה/תיעוד של מבנה הפלטפורמה הפיזית
    - ניתן להגדיר/לעדכן את הפלטפורמה הפיזית גם בשלבים מאוחרים יותר
      - כדאי לעשות זאת בשלב זה כאשר הפלטפורמה הפיזית קיימת / מוכתבת מראש
  - קיימת הקצאה פיזית מובהקת/עקרונית של יכולות המערכת
- תרומת הנדסת המערכת
  - הגדרת/קביעת "קופסאות" החומרה (מחשבים, התקני איחסון, התקנים היקפיים)
  - הגדרת הממשקים הפיזיים (חיבורים, תווכים) – ICD מערכת
  - פריסה (deployment) של התוכנה על פני החומרה
- תרומת הנדסת התוכנה
  - architecture patterns
    - למשל Client/Server, SOA
  - פרוטוקולים
  - ההשלכות התוכנתיות של פתרונות ארכיטקטוניים פיזיים
    - למשל גיבוי, איזון עומסים וכו' במקרה של ריבוי/יתירות של מרכיבים
    - תקשורת ופרוטוקולים במקרה של client-server

©Prof. Amir Tomer

Deployment Model: מודל להגדרת המבנה הפיזי



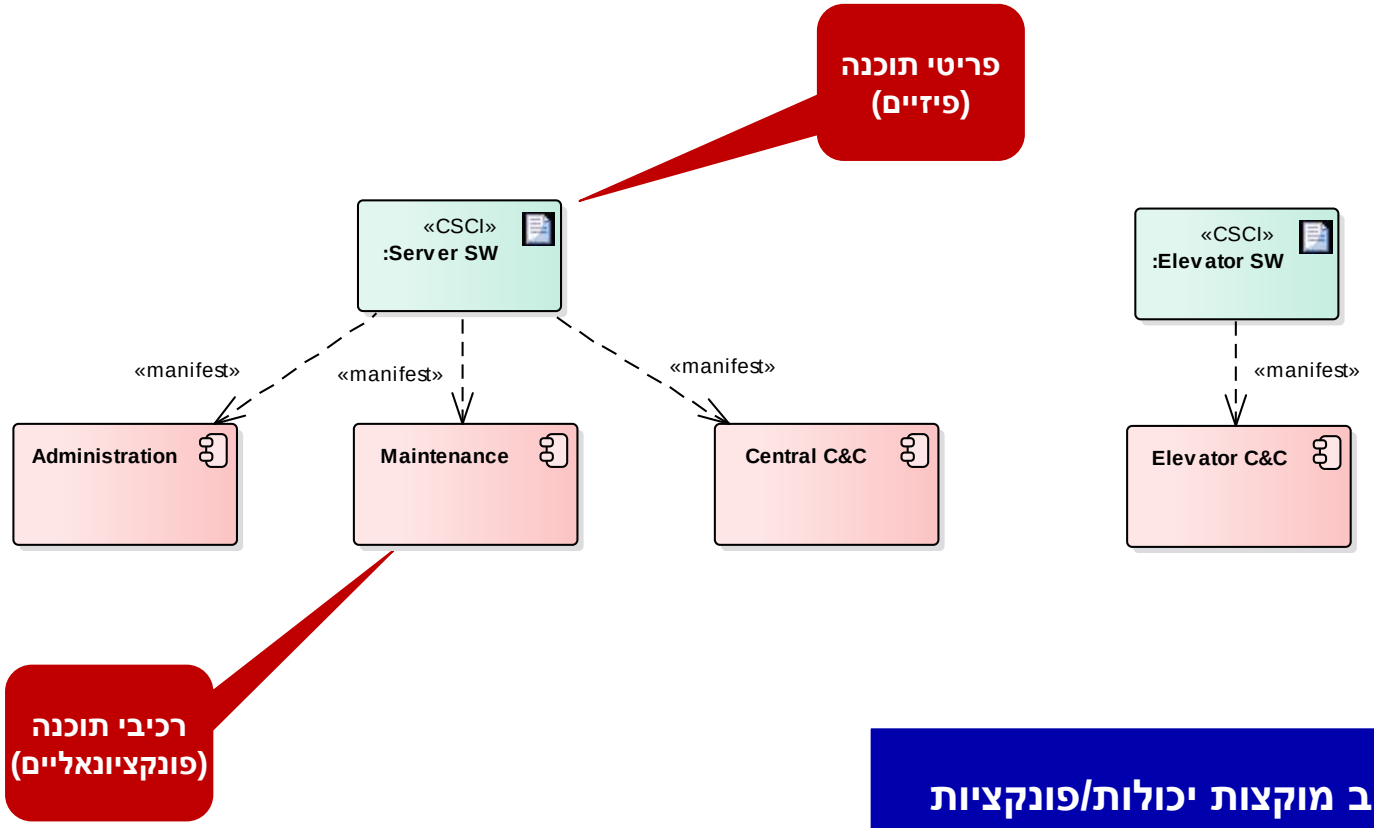
©Prof. Amir Tomer

אנליזה פונקציונאלית

- מהות הפעילות
  - מיצוי הפונקציונאליות הנדרשת מתוך התהליכים (יכולות, פונקציות)
  - הקבצת לקבוצות של פונקציות
  - הקצאת הפונקציות לרכיבים פונקציונאליים (קיימים או חדשים), תוך שמירה על עקרונות של
    - לכידות הדוקה (tight cohesion): הרציונאל שמאחורי הקצאת פונקציה X דווקא לרכיב Y
    - צמידות רופפת (loose coupling): היכולת של רכיב X לשתף פעולה עם רכיב Y מבלי להכיר את המימוש שלו
- תרומת הנדסת המערכת
  - זיהוי הפונקציות המערכתיות (יכולות), כולל אלה הנובעות ממאפייני איכות (למשל security)
  - אילוצי איחוד/הפרדה של יכולות
  - חלוקת ההקצאה של מאפייני איכות
- תרומת הנדסת התוכנה
  - משמעויות האיחוד/ההפרדה של יכולות (למשל כושר חישוב, קיבולת זכרון, מבני נתונים לעומת בסיס נתונים)
  - שימוש ברכיבים קיימים: Open source, ספריות, שירותי מערכת הפעלה
  - חלופות בין רכיבים במימוש (כגון ספריות סטטיות) לבין רכיבים בשימוש (כגון ספריות דינאמיות – DLL)

©Prof. Amir Tomer

manifest: מודל להקבצת פונקציות והקצאה לרכיבים



לכל רכיב מוקצות יכולות/פונקציות שמוצו מתהליכי המערכת בפעילות האנליזה הפונקציונאלית

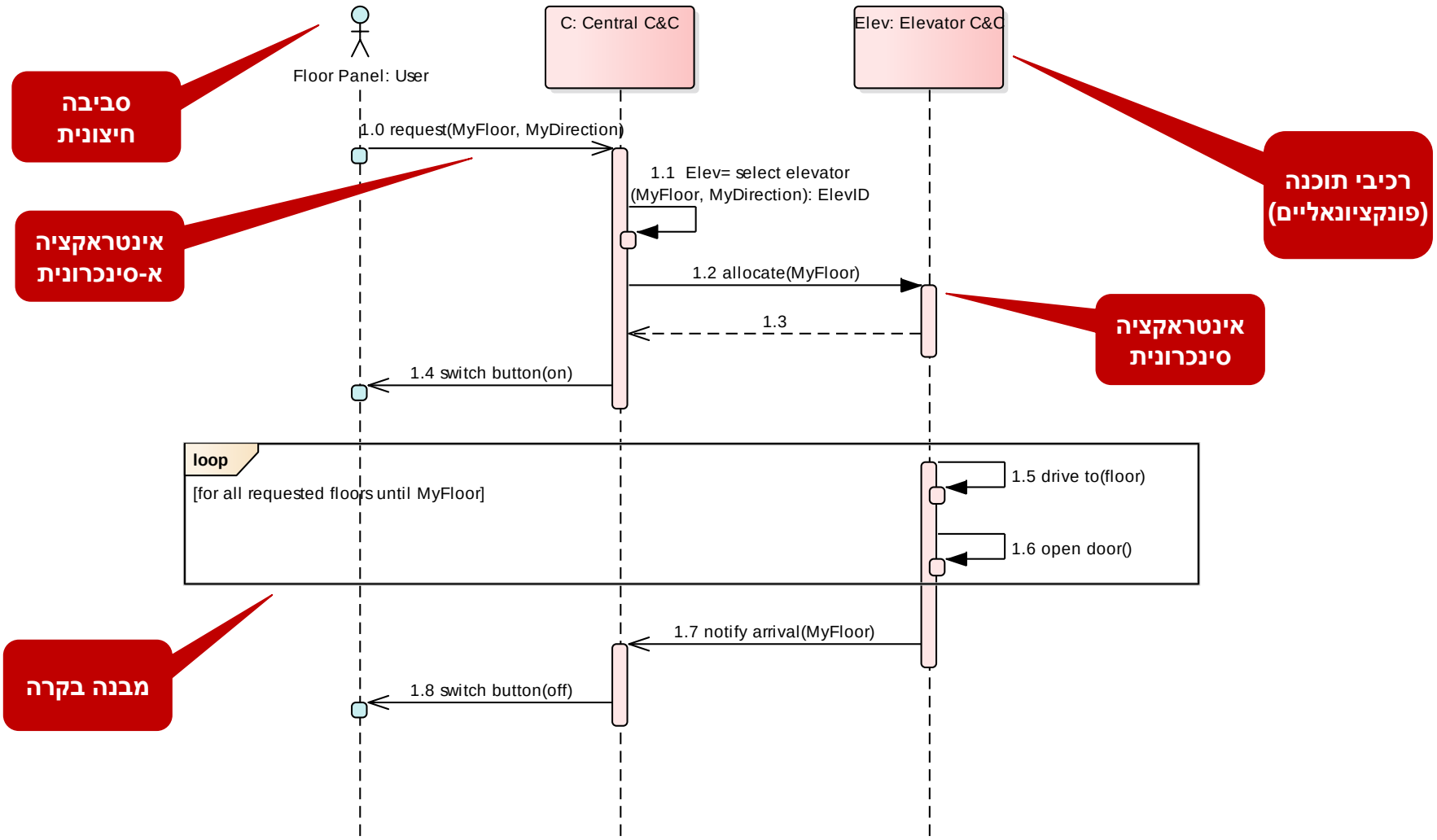
©Prof. Amir Tomer

הגדרת תהליכים פונקציונאליים

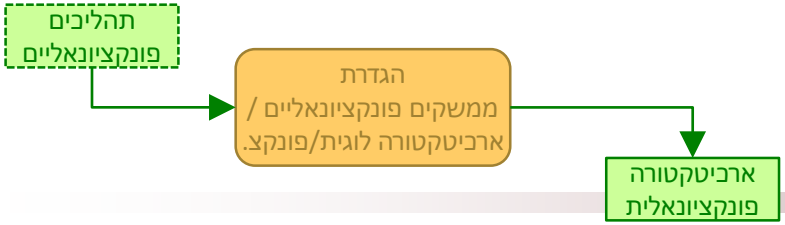
- מהות הפעילות
  - תהליך פונקציונאלי = מימוש של תרחיש בתהליך מערכתי באמצעות אינטראקציה בין הרכיבים הפונקציונאליים
  - התהליכים הפונקציונאליים ימומשו בהמשך באמצעות תוכנה
- תרומת הנדסת המערכת
  - הגדרת האינטראקציה הפונקציונאלית / חלוקת העבודה בין הרכיבים
  - החלטות לגבי אינטראקציה סינכרונית / א-סינכרונית
- תרומת הנדסת התוכנה
  - אינטראקציה עם רכיבים "תוכנתיים" (בסיס נתונים, מתווכים, רכיבי גיבוי)
  - משמעויות של סינכרון / אי-סינכרון פונקציונאלי בין רכיבים
  - Software Architecture Patterns
    - בגון MVC, peer-to-peer, pub/sub

©Prof. Amir Tomer

Sequence Diagram: מודל לתהליכים פונקציונאליים



©Prof. Amir Tomer

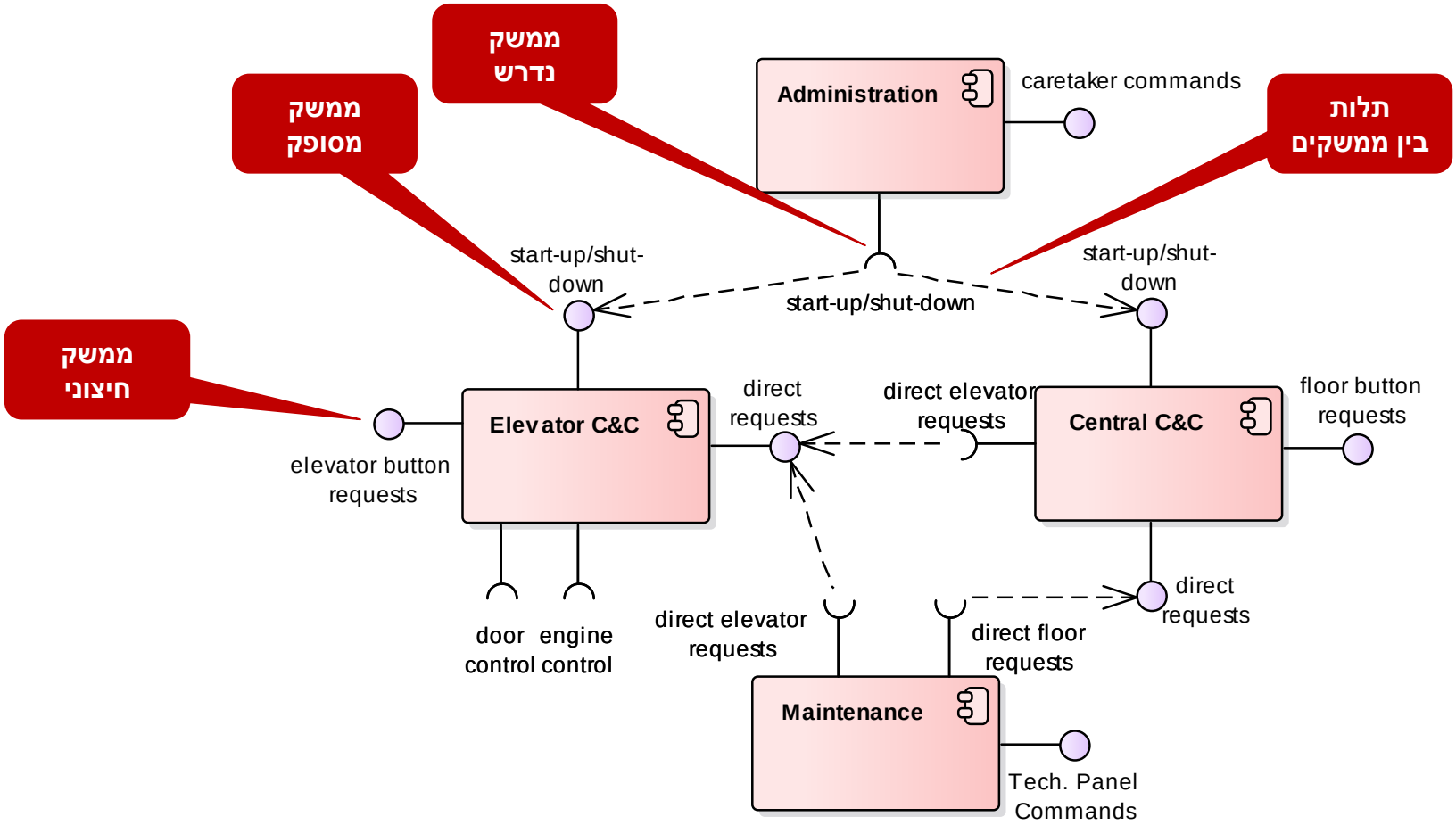


הגדרת ממשקים פונקציונאליים / ארכיטקטורה לוגית/פונקציונאלית

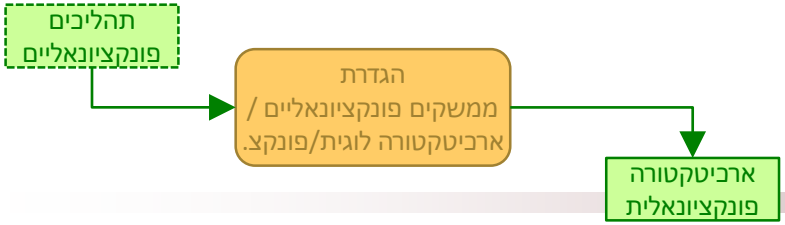
- מהות הפעילות
  - הגדרת הממשקים הפונקציונאליים של כל רכיב, על בסיס האינטראקציה שלו
    - ממשקים מסופקים: הממשקים אותם חושף הרכיב לסביבתו, לצורך פניה אליו
    - ממשקים נדרשים: ממשקים אותם נדרש הרכיב ליישם לצורך פניה לרכיבים אחרים
  - בניית ארכיטקטורה לוגית/פונקציונאלית ע"י קישור בין ממשקים נדרשים למסופקים
- תרומת הנדסת המערכת
  - הקצאת האחריות לממשק (מסופק/נדרש)
  - הגדרת API / הודעות בממשקים cross-platform
- תרומת הנדסת התוכנה
  - הגדרת סוגי הממשקים
    - ישיר (קריאה למתודה)
    - פנימי (בין תהליכים)
    - חיצוני (באמצעות הודעות)
  - הגדרת ICD תוכנה

©Prof. Amir Tomer

Component Model: מודל לארכ. לוגית / פונקציונאלית



©Prof. Amir Tomer

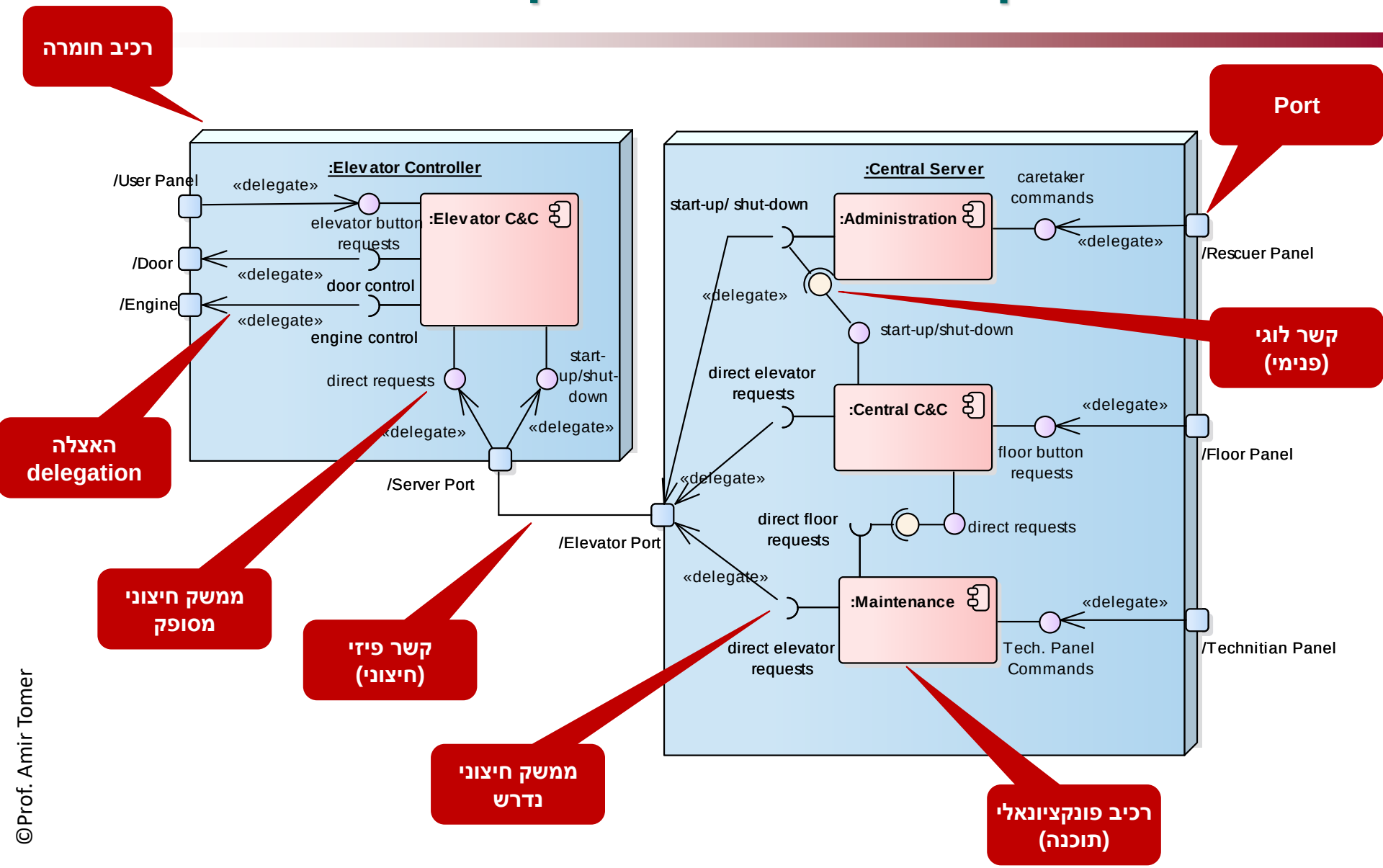


בניית ארכיטקטורה משולבת

- מהות הפעילות
  - הגדרת הארכיטקטורה המשולבת חומרה/תוכנה
  - אינטגרציה של המודלים / התהליכים הקודמים
- תרומת הנדסת המערכת
  - הגדרת ports לממשקים הפיזיים (בכל רכיב חומרה)
  - מיפוי ממשקים לוגיים לפיזיים (האצלה – delegation)
  - אין ממשקים פיזיים מיותרים
- תרומת הנדסת התוכנה
  - וידוא ביסוי
    - לכל ממשק לוגי חיצוני יש מיפוי (האצלה) לממשק פיזי מתאים (דרך port)

©Prof. Amir Tomer

Composite Model: מודל לארכיטקטורה משולבת



©Prof. Amir Tomer

סיכום ומסקנות

- תהליך בניית הארכיטקטורה המערכתית הוא מאמץ משותף של הנדסת המערכת והנדסת התוכנה
- התהליך מתנהל במתוכנת של "מרוץ שליחים"
  - מטרה משותפת
  - תוצרים משותפים
  - תרומה ייחודית של כל דיסציפלינה
- תהליך בניית הארכיטקטורה הוא תהליך מתודולוגי ואיטרטיבי
- בניית ארכיטקטורה מבוססת-מודלים יבולה לשפר את התקשורת בין הצדדים ואת מובנות התוצרים

©Prof. Amir Tomer

סדנאות בחודש יולי

- כתיבת דרישות ותרחישים למערכת עתירת תוכנה
  - משך: יום אחד
  - קהל יעד: מהנדסי מערכות עתירות-תוכנה, ארכיטקטי תוכנה, מהנדסי תוכנה, מהנדסי בדיקות
  - תועלות עיקריות: כתיבה נכונה של דרישות ותרחישים
- הנדסת תוכנה למהנדסי מערכות
  - משך: יומיים (רצופים)
  - קהל יעד: מהנדסי מערכות שאינם ארכיטקטי תוכנה
  - תועלות עיקריות: הבנת היבטי תוכנה במערכת רב-תחומית והשתתפות בתהליך בניית הארכיטקטורה המשולבת
- ארכיטקטורה של מערכות מורכבות עתירות תוכנה
  - משך: 3 ימים (יום בשבוע)
  - קהל היעד: ארכיטקטי תוכנה
  - תועלת עיקרית: יישום טקטיקות תוכנה למאפייני איכות מערכתיים (ביצועים, זמינות, אבטחה וכו'), מעבר מארכיטקטורה פונקציונאלית לארכיטקטורה על בסיס מאפייני איכות (דרישות לא-פונקציונאליות), יישום של architecture patterns

הערה: קיימת חפיפה בין תכני הסדנאות. מומלץ לכל אחד לבחור סדנה אחת המתאימה לו ביותר

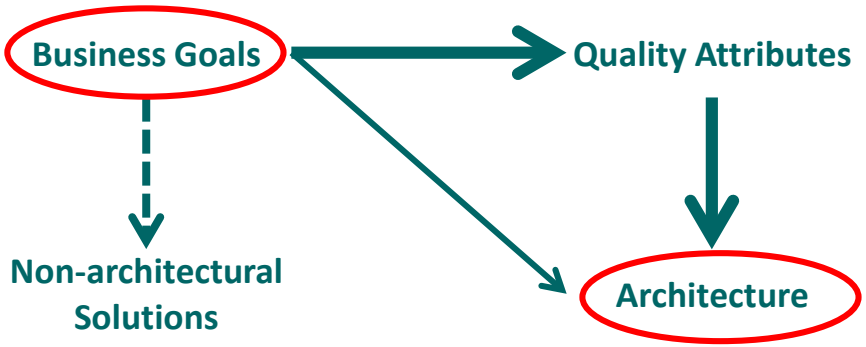
©Prof. Amir Tomer

כאן סיימנו להיום.

©Prof. Amir Tomer

Architecture Significant Requirements (ASR)

- דרישות ASR הן דרישות מפורשות או נגזרות אשר עשויה להיות להן **השפעה משמעותית** על הארכיטקטורה
  - השפעה על היעדים העסקיים, כלומר, דרישה זו הינה חיונית לצורך...
    - שיפור חווית המשתמש
    - הגנה על פרטיות השימוש/המידע של הלקוח
    - השגת יתרון על פני מתחרים
    - הוזלת המחיר
    - ...
  - השפעה על הארכיטקטורה, כלומר, לצורך עמידה בדרישה זו יש צורך ב...
    - הגדלת משאבים (מיחשוב, איחסון, ערוצי תקשורת, ...)
    - ביזור/הפרדה של חלקים במערכת (למשל דרישה להפרדה פיזית מטעמי בטחון)
    - שינויים במנגנוני תקשורת
    - הוספת רבדים (הגנה, תרגום וכו')
    - ...



©Prof. Amir Tomer