

ארכיטקטורה למימוש רשת שירותים

ארגונית המבוססת על ה-Grid

חיבור לשם קבלת התואר "דוקטור לפילוסופיה"

מאת :

רון יוכפז

בית-הספר למנהל עסקים

הוגש לסנט של אוניברסיטת בר-אילן

אב תש"ע

רמת גן

ארכיטקטורה למימוש רשת שירותים

ארגונית המבוססת על ה-Grid

חיבור לשם קבלת התואר "דוקטור לפילוסופיה"

מאת :

רון יוכפז

בית-הספר למנהל עסקים

הוגש לסנט של אוניברסיטת בר-אילן

אב תש"ע

רמת גן

עבודה זו נעשתה בהדרכתו המסורה של פרופסור
דיוויד שוורץ מבית-הספר למנהל עסקים של
אוניברסיטת בר-אילן.

תוכן עניינים

1.....	מבוא	1.
2.....	רקע	2.
4.....	2.1 הגדרת הצורך	
6.....	סקירה תיאורטית ועבודות רלוונטיות	3.
6.....	3.1 כללי	
6.....	3.2 רעיון מסדר לשירותים הארגוניים	
7.....	3.3 עלייתם של פתרונות האינטגרציה	
8.....	3.4 ה-SOA כתשתית אינטגרציה מלאה והופעת ה-ESB	
9.....	3.5 ה-ESB המבוסס על ה-Web Services וחסרונותיו	
12.....	3.6 הבנת הדרישות האוטונומיות של הפלטפורמה וביזורה	
13.....	3.7 ההתפתחות והפתרונות בעולם ה-Grid	
17.....	חזון ה-ESB	4.
17.....	4.1 התפיסה	
18.....	4.2 שכבות מרכזיות בארכיטקטורת ה-SOA	
20.....	4.3 הגדרת ה-ESB	
21.....	4.4 מרכיבי ה-ESB	
24.....	4.5 תסריטי שימוש ל-ESB	
25.....	4.6 המאפיינים העיקריים של ESB ארגוני	
26.....	4.7 מגבלות ה-ESB במצב הקיים	
27.....	4.8 ה-ESB המבוסס על ה-Grid	
29.....	ארכיטקטורת ה-OGSA	5.
29.....	5.1 הרעיון המסדר	
30.....	5.1.1 תקן OGSA והקשר לעולם ה-SOA	
30.....	5.1.2 תקן OGSA - רקע כללי	
31.....	5.1.3 יכולות הליבה בארכיטקטורת ה-OGSA	
32.....	5.2 ייעוד הארכיטקטורה	
35.....	5.3 קישוריות ותמיכה בסביבות הטרוגניות דינמיות	
36.....	5.4 שיתוף משאבים בין-ארגוניים	
37.....	5.5 אופטימיזציה (Optimization)	
37.....	5.6 הבטחת רמת השירות (QoS)	
38.....	5.7 הפעלת גיובים (Jobs Execution)	
39.....	5.8 שירותי הנתונים והמידע (Data Services)	
39.....	5.9 אבטחת המידע (Security)	
41.....	5.10 הפחתת עלויות הניהול (Administrative Cost Reduction)	
41.....	5.11 יכולת גידול (Scalability)	
42.....	5.12 זמינות (Availability)	
43.....	5.13 נוחות השימוש והתרחבות (Ease of Use and Extensibility)	
44.....	מיפוי יכולות ה-ESB ליכולות ה-OGSA	6.
47.....	ארכיטקטורת ESB מבוססת על בסיס ה-OGSA	7.
50.....	מתודולוגיה, כלי הניתוח וחומרים	8.
50.....	8.1 חלופות	
50.....	8.2 מאפיינים איכותיים להשוואה בין החלופות	

51	8.3. כלי לניתוח והשוואה	
51	8.4. מודל הבחינה וההשוואה	
51	8.5. מודל ה-UCM	
52	8.6. מודל ההשוואה – ATAM	
53	8.7. שימוש במודל ATAM	
54	8.8. מאפיינים איכותיים להשוואה	
55	8.9. הגדרת נקודות רגישות ונקודות Tradeoff	
56	8.10. תרחישי השימוש (Use Cases) במודל ההשוואה	
58	השוואה ותוצאות	9.
58	9.1. תרחישי ההשוואה	
58	9.1.1. שירותי ביטחון מערכות המידע - הזדהות והרשאות	
61	9.1.2. שירותי שינוע המסרים - התמנות לאירוע	
63	9.1.3. שירותי אירוח השירותים - טרנזקציות ואישור האירוע	
65	9.1.4. שירותי ממשק משתמש-תצוגה	
67	9.1.5. שירותי אחסון המידע - ניהול האחסון	
69	9.1.6. שירותי שליטה ובקרה (ש"ב) - ניטור מצב הרכבים	
70	9.1.7. שירותי הגילוי לתחנות המוניות	
72	9.1.8. שירותי שיתוף - Virtual Organizations	
73	9.1.9. שירותי תיווך - המרת הנתונים	
75	9.2. השוואת התרחישים על בסיס פרמטרים הנבחנים ב-ATAM	
75	9.2.1. כללי	
75	9.2.2. אבטחת המידע	
76	9.2.3. זמינות השירות	
76	9.2.4. גמישות לשינויים	
76	9.2.5. עץ השימושיות	
77	9.3. סיכום ההשוואה	
79	10. סיכום ומסקנות	
79	10.1. עיקרי המסקנות	
79	10.2. פירוט המסקנות	
80	10.2.1. השוואת הממצאים של Grid Service לעומת אלה של Web Service	
82	10.3. נושאים למחקר עתידי	
83	10.3.1. המלצות לשימוש בשירותים מבוססי Grid	
83	10.3.1.1. וירטואליזציה	
84	10.3.1.2. סטנדרטיזציה	
84	10.3.1.3. אינטגרציה ושיתוף באמצעות Virtual Organization	
84	10.3.1.4. תמיכה ב-Stateful Web Services	
85	10.4. תרומת המחקר	
79	11. ביבליוגרפיה	
90	נספח א – סקירת פרדיגמת ה-SOA (Service Oriented Architecture)	
91	כיצד התבצעה האבולוציה של SOA	
92	מרכיבי ארכיטקטורת SOA	
95	מאפייני SOA	
95	היתרונות של אימוץ SOA	
97	נספח ב – סקירת נושא ה-ESB (Enterprise Service Bus)	
101	נספח ג – מחשוב Grid - Grid Computing	
101	ארכיטקטורת OGSA (Open Grid Service Architecture)	

102	שכבות ה-OGSA
104	נספח ד – יכולות ארכיטקטורת OGSA
104	1. כללי
106	2. תיחום OGSA (OGSA Framework)
108	3. שירותי תשתית (Infrastructure Services)
110	4. ניהול הפעלת השירותים (Execution Management Services)
120	5. שירותי גישה לנתונים (Data Services)
129	6. שירותי ניהול משאבים (Resource Management Services)
133	7. שירותי אבטחה (Security Services)
143	8. שירותים בניהול עצמי (Self-Management Services)
149	9. שירותי מידע (Information Services)
155	נספח ה – דוגמא לפעילות מחקרית בנושא ה-SOA וה-Grid

רשימת הטבלאות

27	טבלה 1 - מאפייני התשתית On Demand Information Environment
32	טבלה 2 - יכולות הליבה של ארכיטקטורת ה-OGSA [15]
35	טבלה 3 - דוגמאות לתסריטי השימוש [15]
45	טבלה 4 - כיסוי משפחות השירותים על-ידי OGSA
46	טבלה 5 - כיסוי משפחות השירותים על-ידי רכיבי ה-GT4
57	טבלה 6 - תרחישי השימוש
78	טבלה 7 - סיכום השוואת התרחישים
130	טבלה 8 - יחסיות בין יכולות ניהול לממשקים

רשימת איורים

2	איור 1 - ארכיטקטורת ה-SOA [12]
4	איור 2 - שכבות רשת השירותים [24]
5	איור 3 - מעבר מ-ESB מבוסס BUS ל-ESB מבוסס Grid
11	איור 4 - ארכיטקטורת ה-xSOA [1]
16	איור 5 - איחוד סטנדרטי של ה-Web Services וה-Grid בדרך לתקן מרכזי WS-I
17	איור 6 - מבנה ייחוס שלם לארכיטקטורת ה-SOA [24]
19	איור 7 - מבנה ה-Bus Business Service [24]
23	איור 8 - שירותי התשתית של ה-GIG [22]
24	איור 9 - תסריט שימוש בסיסי
28	איור 10 - הדגמה ויזואלית לתפיסת רשת השירותים [24]
31	איור 11 - יכולות ארכיטקטורת ה-OGSA [15]
46	איור 12 - החפיפה בין ה-GIG, Globus והסטנדרטים של Web Services [32]
47	איור 13 - Grid Service (העוטף את ה-Service)
48	איור 14 - שני Grid Services הרצים על Web Server אחד
49	איור 15 - דוגמה לשרת המכיל כמה שירותים הניתנים על-ידי OGSA
49	איור 16 - תהליך מציאת השירות ב-ESB על בסיס תשתיות ה-Grid
58	איור 17 - מימוש הזדהות ללא Token
59	איור 18 - מימוש ההזדהות באמצעות Grid token
60	איור 19 - שימוש בשירות הישירות (המשתמש מבצע את פעולות ההצפנה)

איור 20 - קבלת מידע על אירוע באופן סנכרוני (שליחת שאילתה כל כמה זמן)	62
איור 21 - התמנות לאירוע וקבלת ההתראה במהלך התרחשות האירוע – מספר הפניות נמוך משמעותית	63
איור 22 - מימוש ההזמנה בהתבסס על שירות ההתמנות	64
איור 23 - מימוש ההזמנה וביצוע הטרוקציה ב-Grid	65
איור 24 - מימוש התצוגה באמצעות שירותי המרת הנתונים	66
איור 25 - מימוש התצוגה ללא תלות במקור המידע	66
איור 26 - מימוש אחסון עצמאי	67
איור 27 - אחסון באמצעות MDS4 ו-OGSA DAI	68
איור 28 - מציאת הנהג המיטבי באמצעות נתוני שו"ב	69
איור 29 - מימוש מציאת הנהג המיטבי באמצעות אופטימיזציות	70
איור 30 - תהליך הרישום וגילוי סטטי	71
איור 31 - תהליך גילוי דינמי	72
איור 32 - מימוש "VO" הדורש רכיב ניהול מרכזי ומוסכם	73
איור 33 - יצירת VO באופן דינמי וללא תלויות	73
איור 34 - המרת הנתונים באמצעות שירות תיווך ייעודי	74
איור 35 - המרת הנתונים באמצעות שירות מובנה	74
איור 36 - ארכיטקטורת שירותים בסיסית [1]	90
איור 37 - אפשרויות מימוש לארכיטקטורת ה-SOA	92
איור 38 - מרכיבי ארכיטקטורת ה-SOA	93
איור 39 - אופן פעולת ה Proxy בארכיטקטורת השירותים	94
איור 40 - סכימה כללית למבנה ESB [24]	97
איור 41 - ארכיטקטורת Service Oriented Business Application [24]	98
איור 42 - שיתוף עם פתרונות קיימים (ESB Co-Exists with Existing Middleware) [24]	99
איור 43 - שכבות רשת השירותים [24]	99
איור 44 - וירטואליזציה של משאבים תשתיתיים ועסקיים [24]	100
איור 45 - ארכיטקטורת OGSA [15]	102
איור 46 - תרשים קונספטואלי של תשתית Grid [15]	105
איור 47 - OGSA framework [15]	107
איור 48 - מיפוי היחסים בין השירותים [15]	108
איור 49 - EMS מטה דטה מחולק לעולם המספק ולעולם הדורש [15]	112
איור 50 - שימוש בשירותי EMS להפעלת Legacy Job [15]	120
איור 51 - ניהול המידע והמטה דטה של עולם הצרכן ועולם הספק [15]	123
איור 52 - רמות ניהול ב OGSA [15]	131
איור 53 - שיתופיות בין ארגונית [15]	135
איור 54 - דוגמא לקלט עבור מדיניות החלטה ואכיפה [15]	136
איור 55 - רעיון הקהילה הוירטואלית [15]	137
איור 56 - ספריה דיגיטאלית כדוגמא לקהילה וירטואלית [15]	138
איור 57 - פריסת הרשאות [15]	138
איור 58 - שירותי לוג מאובטח בסביבה מבוזרת [15]	139
איור 59 - שירותי אבטחה בסביבה מבוזרת [15]	140
איור 60 - מיפוי הרכיבים במודל אבטחה של Grid [15]	142
איור 61 - דוגמא לתרחישים : שכבת Grid ושכבת ה Job [15]	145

תקציר

בשנים האחרונות אנו עדים לזליגה של ארכיטקטורות וטכנולוגיות מעולם האינטרנט אל עולם המחשוב הארגוני, כגון טכנולוגיות Web Services. ארכיטקטורות וטכנולוגיות אלו אשר הגיעו מעולם האינטרנט, חדרו אל המחשוב הארגוני ואפשרו את פיתוח תפיסות העבודה החדשות המבוססות על קונספט של צריכת שירותים דינמית, כגון SOA - Service Oriented Architecture.

מחקר זה מצביע על מגמה הפוכה של זליגה בארכיטקטורות ובתפיסות, שעד כה זכו למימוש בגבולות הארגון, כגון טכנולוגיות ה-Grid וארכיטקטורה המבוססת על שירותים, אל כיוון האינטרנט ופיתוחי ה-Web. ארכיטקטורות אלו, מאפשרות חשיפה וצריכה דינאמית של יכולות המחשוב והאפליקציה מספקי שירות פנימיים וחיצוניים לארגון.

המחקר מציג את היכולת לייצר פלטפורמת שירותים חדשה ומבוצרת, המאפשרת צריכת שירותים, בצורה מאובטחת וסינכרונית, ממקורות שונים בארגון ואף מחוץ לארגון. לשם כך המחקר מתמקד בחדשנות מחשוב ה-Grid והתאמתו למימוש דרישות פרדיגמת השירותים של עולם ה-SOA, ושכבת תשתית ה-Enterprise Service Bus (ESB).

המחקר מציג את תפיסת ה-ESB, כתשתית מרכזית לחשיפת השירותים בעידן ה-SOA ולצריכתם, תוך התבססות על מחקר מקיף של משרד ההגנה וצבא ארה"ב בתחום. ההגדרות שנעשו על ידי צבא ארה"ב, מגדירות תשע משפחות שירותי בסיס, המהוות אפיון של יכולות הליבה הנדרשות לתשתית מבוצרת זו.

בשלב הראשון המחקר מציג ארכיטקטורה חדשה המבוססת על ה-Grid ליישום פלטפורמת תשתית השירותים (תשע משפחות שירותי הבסיס), אשר תאפשר את חשיפתם באופן סטנדרטי על גבי הרשת, ואת צריכתם. כל זה ללא צורך במתווך מרכזי ובאופן מנוהל ומבוצר לחלוטין (Grid based ESB). ארכיטקטורה זו נסמכת על החדשנות של טכנולוגיות ה-Grid וליישום ה-Grid Services ובאה לענות על הבעיות של הארכיטקטורה הקיימת כיום, קרי פתרונות ESB המבוססים על Web Services, ובעיקר על נושא הביזור והדינמיות, אשר מהוות מחסום לגידול בפתרונות הקיימים.

בשלב השני בוצעה הגדרה של תרחישי השימוש והמאפיינים האיכותיים, לשם השוואה איכותית בין הארכיטקטורות, המבוססת על המתודולוגית Tradeoff Analysis Model (מודל להשוואה איכותית של הארכיטקטורות). זאת, על מנת לבחון את היכולות של הארכיטקטורה החדשה, בעיקר מול מגבלותיה של הארכיטקטורה הקיימת. תוצאות ההשוואה מראות את יתרונות הארכיטקטורה החדשה בהתמודדות עם חשיפת השירותים בקני מידה גדולים, תוך שמירה על דינמיות ויכולת עבודה מבוצרת, ללא התפשרות על נושאי אבטחת המידע וצריכתם.

המחקר, המציג תפיסה חדשנית, הינו חלק מתחילתו של עידן מחשוב חדש, שלאחרונה אנו חווים את תחילתו תחת הכותרת של שירותי מחשוב הענן (Cloud Computing). שירותים אלה הם, למעשה, הדגמה של תפיסת ה-Grid, החוצה את הגבולות הפנימיים והחיצוניים של הארגון והמאפשרת צריכה של יכולות ושירותים באופן מבוזר ודינמי. הארכיטקטורה והתשתית המוצעים במחקר, יחד עם המלצות בתחומי הוירטואליזציה, הסטנדרטיזציה והשיתוף, מהווים נדבך חשוב ומקיף להמשך התפתחותו של מחשוב בתפיסת הענן. התשתית הארכיטקטונית המוצעת מקדמת את יכולות המחשוב שצרכני השירותים וספקיהם נדרשים אליהם, על מנת לספק שירותים אפליקטיביים בעידן הענן.

1. מבוא

התפתחות תחום מחשוב ה-Grid ברמה האפליקטיבית בשנים האחרונות הביאה לגל מחקרי חדש ולהתעוררותן של שאלות מחקר אקדמיות רבות [6,14]. אחת השאלות דנה ביחס לחדשנות מחשוב זה והתאמתו למימוש דרישות פרדיגמת ה-SOA [1,15]. לשם כך יש לבחון מהן הגדרות ודרכי המימוש הנדרשות לשם התאמת ארכיטקטורת ה-Grid המבוזרת לפרדיגמת ה-SOA, ויכולת מימוש ה-ESB המבוזר. מחקר זה מציע ארכיטקטורה חדשה למימוש רשת השירותים המבוזרת על בסיס טכנולוגיית ה-Grid ומוכיחה את יעילותו ואת התאמתו לחזון ה-SOA, באמצעות בחינה והשוואה לארכיטקטורות הקיימות. תוצאות עבודה זו מניחות בסיס לתשתית שירותים מבוזרת, המאפשרת חשיפה וצריכה יעילה של שירותים בצורה דינמית, מאובטחת וסקלבילית.

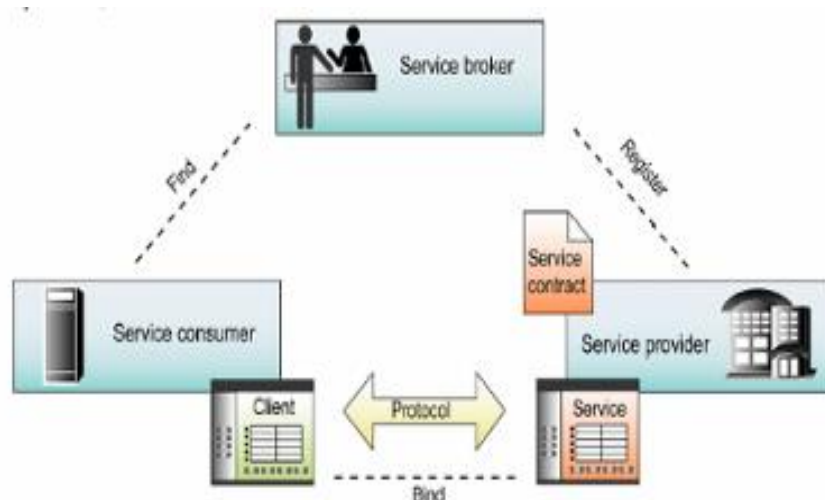
מבנה העבודה :

- בפרקים 2 ו-3 ניתן מבוא וסקירה תיאורטית של המחקרים שעליהם מושתת עבודה זו.
- בפרקים 4 ו-5 נגבש את ההגדרות והחדשנות הטכנולוגית בבסיס ההצעה לגיבוש ארכיטקטורה חדשה עבור רשת שירותים מבוזרת.
- בפרקים 6 ו-7 נציג את הארכיטקטורה המבוזרת.
- בפרק 8 נציג את מודל הבחינה לארכיטקטורה המוצעת, כאשר בחינת יכולותיה והשוואתן לארכיטקטורות הקיימות תתבצע בפרק 9.
- פרק 10 יסכם את המחקר וייספק תחזיות להמשך.

2. רקע

התפתחות עולם האינטרנט ורשת ה-Web הביאה עמה את תפיסת השירותים האפליקטיביים ברשת (Web Services), שיצרה הזדמנויות עסקיות וטכנולוגיות חדשות והביאה לידי חיסכון בפיתוח כפול של אפליקציות דומות. שירותים אלה, שפעלו ברשת, אפשרו פשטות, חיסכון ודינמיות רבה יותר בצריכתם ויצרו מהפכה באופן החשיבה והפיתוח של המוצרים והאפליקציות החדשות, אשר באו לידי ביטוי במהרה ברשת העולמית והארגונית. כך, התפסה כי אפשר להתבסס על הרשת כמקור ארגוני משותף לצריכת השירותים הבשילה ואף כמקור המאפשר חשיפה וצריכה אוטונומית שלהם.

אבולוציה זו הביאה להבנה שאפשר לתפוס את הרשת כ"ארכיטקטורה מכוונת שירותים" – SOA – Service Oriented Architecture כרעיון מסדר וחסכוני לפיתוח האפליקטיבי הארגוני. העיקרון הבסיסי שעליו מושתתת הבנה זו הוא שכל שירות יכול לחשוף את עצמו ברשת, להירשם בקטלוג משותף (Publish), להיחשף לשירותים האחרים (Find) ולצרוך אותם בצורה אוטונומית (Bind) ומנוהלת [1], כפי שאפשר לראות באיור 1.



איור 1 - ארכיטקטורת ה-SOA [12]

פרדיגמת ה-SOA מציגה, למעשה, תובנה חדשה, שלפיה אפשר להפסיק עם הפיתוח הוורטיקלי של מערכות מונוליטיות (Monolithic) שלמות ולהחליפן בצריכה מודולרית של כמה שירותים על גבי הרשת. כך מתאפשר איחוד (Aggregation) של כמה שירותים לשירות חדש, אד-הוקי, לטובת משימה מוגדרת (CA – Composite Application) [3]. כך הפרדיגמה מעבירה את כובד המשקל מן המערכות הארגוניות (System Centric) לרשת הארגונית (Net Centric) ומעלה שאלות מחקריות רבות ביחס לשירותים הבסיסים שאליהם היא מחויבת [7]. תפיסת השירותים, כפי שהיא נגזרת מהגדרת ה-SOA, אינה חדשה, אלא פרדיגמה הקיימת מזה כמה שנים המנסה ליצור סדר ושימוש חוזר (Reuse) ביכולות קיימות הנחשפות כשירותים (Services). יחד עם זאת, ה-SOA נשארה בגדר פרדיגמה בלבד, שכן מרבית הארכיטקטורות

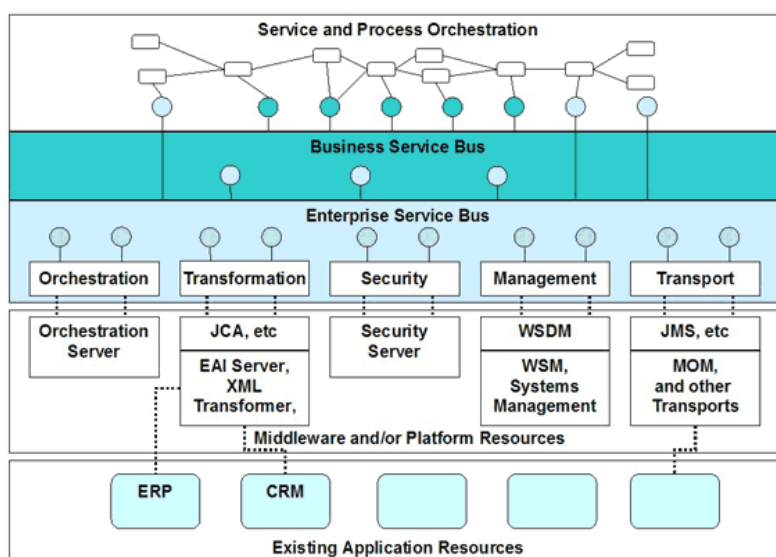
והטכנולוגיות לא יכלו לתת מענה הולם וסטנדרטי לדרישות הכבדות הנגזרות ממנה. ניסיונות המימוש של פרדיגמת ה-SOA באות לידי ביטוי במרבית המחקרים כיום, גם של חברות אינטגרציה בתעשייה וגם בידי גופי פיתוח גדולים המנסים לבדוק מהי הדרך הנכונה לבנות תשתית ארגונית שלמה אשר תתמוך בה? נוסף לכך, כיצד אפשר להפוך את תשתיות האינטגרציה הקיימות לתשתית התומכת בפרדיגמת השירותים, ובכך להשיג גמישות רבה יותר של הארגון (Agility) ולספק את הצרכים העסקיים החדשים והדינמיים המתעוררים בארגון (On Demand Information Environment) [4,16]?

התפיסה המובילה כיום מחלקת את פרדיגמת ה-SOA לשלוש שכבות [24]:

- *Enterprise Service Bus (ESB)* - שכבת התשתית המספקת את שירותי הארגון הבסיסיים (Core Enterprise Services).
- *Business Service Bus (BSB)* - תבנית ארכיטקטורית-לוגית (Logical Architecture Pattern) לארגון קבוצת השירותים העסקיים למרחב עסקי מסוים (Business Domain) לדוגמה: משאבי אנוש.
- *Composite Application (CA)* - יישומים הנבנים על-ידי קומפוזיציות

השירותים העסקיים בשכבת ה-BSB לטובת ביצוע המשימה המוגדרת.

ה-ESB מהווה תבנית ארכיטקטורה לוגית המספקת מבט (Infrastructure View) אל יכולות התשתית הבסיסיות של הארגון. יישום ה-ESB מתבצע תוך הדגשה של שכבת התווך בארגון (Middleware Layer) [4,5]. שכבה זו מספקת את היכולות הנדרשות למימוש תשתית הריצה והניהול, המשמשת את מגוון השירותים המסופקים באמצעות שתי השכבות הנוספות בארכיטקטורת ה-SOA (שירותים עסקיים ושירותים מורכבים). שירותי ה-ESB מהווים, למעשה, תשתית רוחבית, המשמשת את כלל השכבות בארכיטקטורה, כפי שאפשר לראות באיור 2.



איור 2 - שכבות רשת השירותים [24]

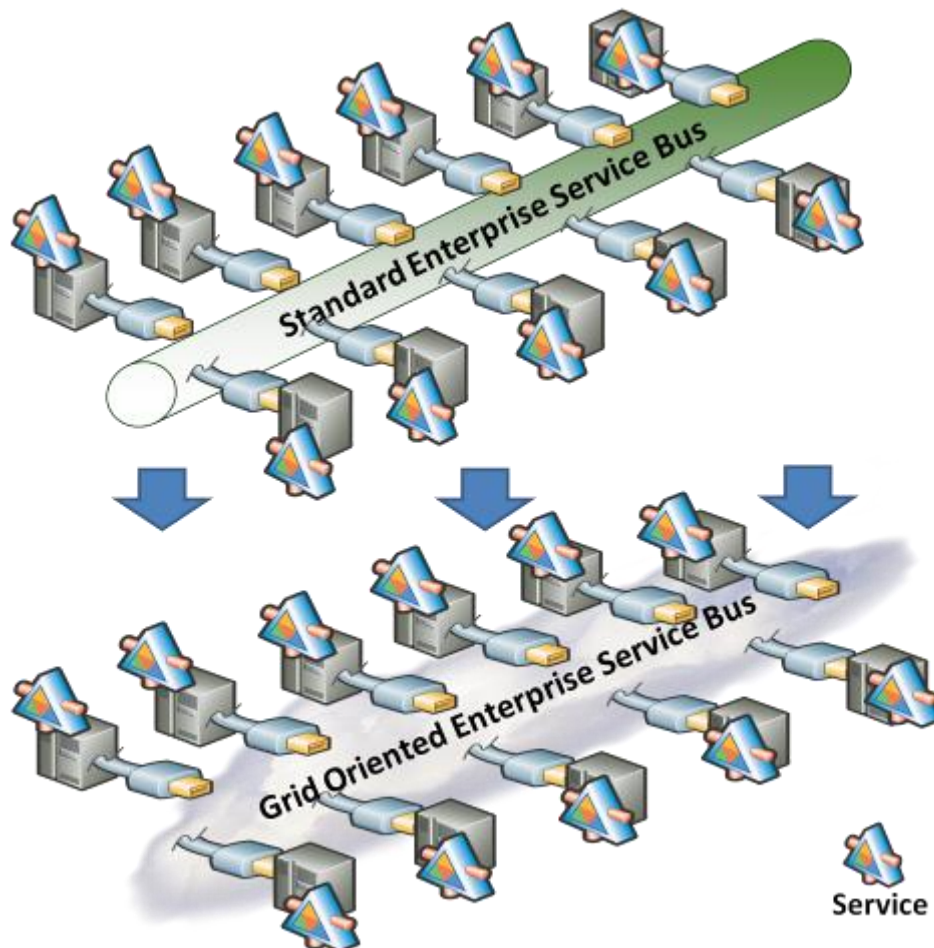
הבעיה המרכזית בפתרונות הקיימים היא שמרבית תשתיות ה-ESB כיום מבוססות על מתווך מרכזי (ברוקר או תשתית bus מרכזית). בעיה זו גורמת גם לעוד כמה בעיות, כגון נקודת כשל מרכזית, יכולת סילום (סקלבליות) מוגבלת ופגיעה בביצועי התשתית הארגונית. פתרון אפשרי מצביע על הצורך בתשתית מתווכת ומנוהלת, שתאפשר מתן **תשתיות שירותים מבוזרות** על גבי הרשת.

התפתחות תחום מחשוב ה-Grid ברמה האפליקטיבית בשנים האחרונות הביאה לגל מחקרי חדש ולהתעוררותן של שאלות מחקר אקדמיות רבות [6,14]. בנוגע לאופן התמודדותה של הארכיטקטורה המבוזרת בעולם ה-Grid האפליקטיבי עם דרישותיה של פרדיגמת ה-SOA [1,15] ושכבת ה-ESB [2,19]. במסגרת מחקר זה נבחן מהן הגדרות המימוש הנדרשות לשם התאמת ארכיטקטורת ה-Grid המבוזרת לפרדיגמת ה-SOA ויכולת מימושו של ה-ESB המבוזר.

2.1. הגדרת הצורך

כיום, יותר מתמיד, קיים הצורך ב"רעיון מסדר", אשר יארגן את משאבי ה-IT של הארגון בעזרת מודל שיספק את היכולת לבנות תשתית שירותים ארגונית. ביכולתה של תשתית זו להסתגל לשינויים התכופים בסביבה העסקית, תוך יצירת רשת משאבים (יישומים ותשתיות), הנרתמים בווקטור אחד למען השגת היעדים העסקיים של הארגון [1,29]. נדרשת, למעשה, תשתית ארגונית גמישה, מותאמת ומנוהלת, שתשמש כפלטפורמת קישוריות וניהול תהליכים בתוך הארגון ובין הארגונים השונים. כל זאת ללא תלות בתשתית תיווך מרכזית (middleware). כל יכולת חדשה שתפותח תיוצג כשירות ברשת העסקית ובעת חיבורה היא תהנה משירותים בסיסיים ברמה האפליקטיבית שהרשת מעניקה, כגון הזדהות, ניהול,

הרשאות גישה, טרנספורמציה בין הנתונים, שירותי מיפוי ועוד - בצורה שקופה ומבוזרת [21]. חשוב לציין כי כלל הפתרונות הקיימים כיום למימוש קונספט ה-ESB לוקים בחסר בכמה תחומים מרכזיים (ייפורט בפרק 4), אשר ימנעו מהם להיות את הבסיס למימוש ה-On Demand Information Environment. מחקר זה מכוון לכך שהפתרון מצוי בהיתוך (Fusion) קונספט ה-Grid עם קונספט ה-ESB וה-SOA. כך נקבל סוג של ESB המבוסס על ה-Grid, אשר ישמש סוג של מערכת הפעלה ארגונית מבוזרת, דינמית וקלה לניהול, אשר תספק את שירותי המחשוב הארגוני על-פי דרישה למשתמשים השונים [4]. מודל זה מציג גישה חדשה בקישוריות ובאינטגרציית מערכות המידע, בדומה לשינוי שנבע מהתפתחות האינטרנט, אך ברמה אחת מעל רמת התקשורת – האפליקציה. לשם כך, המחקר יבחן את דרכי המימוש הנדרשות לשם התאמת ארכיטקטורת ה-Grid המבוזרת לפרדיגמת ה-SOA ויכולת מימוש ה-ESB המבוזר, שיאפשר מעבר לתשתית שירותים מבוזרת כפי שמודגם באיור 3.



איור 3 - מעבר מ-ESB מבוסס BUS ל-ESB מבוזר מבוסס Grid

3. סקירה תיאורטית ועבודות רלוונטיות

3.1. כללי

הבעיה המרכזית שמעסיקה את מרבית הארגונים כיום היא היכולת להתמודד עם השינויים התמידיים. הארגון נדרש להגיב לאירועים רבים ושונים במהירות וביעילות. יש מתאם ברור בין יכולת הארגון להגיב במהירות וביעילות לשינויים בסביבתו לבין מידת הצלחתו לעמוד ביעדיו [4].

ברוב הארגונים, נכון להיום, משאב ה-IT אינו **ערוך** בצורה אופטימלית על מנת להתמודד עם השינויים בסביבה. הפתרונות הקיימים כיום הם בעיקר מבוססי תשתיות ה-Web Services, הלוקים בחסר והסובלים מבעיות של חוסר סקלריות, בעיות התאמה, סמנטיקה ואבטחה. יש צורך **ממשי** בפתרון אינטגרטיבי שיאפשר דינמיות מלאה, תוך הפשטת השימוש בשירותים ושימוש חוזר (Reuse) באלה הקיימים. נדרשת תשתית ארגונית גמישה, נכונה ומותאמת לשימוש כפלטפורמת קישוריות ולניהול תהליכים בתוך הארגון ובין הארגונים [12]. ארכיטקטורת ה-SOA מהווה גישה חדשה לתכנון עולם ה-IT בארגון, כעולם שמספק שירותים בעלי פונקציונאליות מוגדרת ופשוטה [12]. מדובר בפירוק סביבת התוכנה, שמורכבת ממוצרים רבים, לתבניות של שירותים על הרשת, כאשר כל מערכת מספקת שירותים (אחד או יותר) למערכת אחרת. כל אחת מן המערכות אמורה להתממשק אל המערכות האחרות, לקחת נתונים רלוונטיים וליצור מערכת שמספקת שירותים שונים ומגוונים, תוך שימוש בשפה המובנת לכולם (סמנטית). ארכיטקטורת ה-SOA היא למעשה, גישה לארגון מערכות המידע באופן שבו משאבי הנתונים, הלוגיקה העסקית ותשתיות המחשוב נגישים לכולם באמצעות העברת מסרים בין רשתות של ממשקים סטנדרטיים [4]. מדובר במבנה מחשובי, שבו אפליקציית התוכנה מכילה רק את ההיגיון הספציפי למשימתה המידית ומשתמשת בסט של שירותים ברשת, על מנת לבצע משימות כלליות. כך אפשר לממש את רעיון ה-Grid של הקצאת המשאבים והשירותים הכלליים לפי הצורך [4].

3.2. רעיון מסדר לשירותים הארגוניים

שלב ראשון: לשם בחינת המיקום העתידי, שאנו שואפים אליו, יש לקבוע רעיון המסדר את השירותים הארגוניים. תחום זה נסקר בספרות האקדמית באופן מעמיק. אך, עם זאת, יש לציין כי בכל ארגון נעשית התאמה על מנת לעמוד ביעדים העסקיים והאחרים של הארגון ומתוך ראיית סט השירותים ומידת הביזור הנדרשים בו.

תחום שירותי הרשת (Web Services) הופך לתפיסה מעצבת במסחר האלקטרוני ולקישוריות בין-יישומים ובין המערכות ההטרוגניות. תחום זה כולל מספר רב של פרוטוקולים, כדוגמת WSDL, UDDI ו-SOAP, אשר מגובשים בפורומים מקצועיים [9].

עם זאת, אוסף הפרוטוקולים הנ"ל אינו נותן מענה לצרכים המרכזיים הקיימים כיום בתחום השימוש החוזר במשאבי התוכנה: היכולת לבצע שילוב דינמי של שירותים שונים ממקורות שונים וקיימים באמצעות שימוש חוזר, על מנת לפתח מענה מחשובי בזמן קצר ובעלויות נמוכות לצרכים העסקיים של הארגון [3].

ראוי לציין כי יש השתרשות של יוזמות ראשוניות בתחום שילוב השירותים ממקורות שונים, כדוגמת ה-BPEL4WS. אך, יחד עם זאת, יוזמות אלו מציגות יכולות בשלות בנושאים קריטיים לשם מימוש החזון, כדוגמת התחשבות בעלויות השימוש בזמן אמת ובחירת השירותים על-פי SLA.

תחום אחר הנמצא בתנופה ונותן מענה חלקי לצורך שהוצג לפני כן הוא ה-Grid. סביבה טכנולוגית זו מתמקדת במתן מענה גמיש, מאובטח, תוך שילוב של מקורות ומשאבים מסביבות שונות לשם ביצוע המשימות [14].

שילובם של שני תחומים אלה לשם השגת מטרת-העל מקבל תוקף במסגרת ה-Open Grid (OGSA) Services Architecture, הסטנדרט המעשי בתחום התווכה של Grid Middleware [15].

החיסרון המרכזי של ה-OGSA נגזר מהתפיסה האקדמית, אשר מתמקדת בבניית ה-Grid המאובטח לשם ביצוע חישובים מדעיים וגם במתן מענה לשירותים המסחריים, אשר לא נבנו לצורך המחשוב המקבילי.

כפי שמורחב בסקירה הטכנולוגית על מחשוב ה-Grid (נספח ד') ותחום ה-OGSA, נראה כי הדור הבא של הפתרון המשולב בין מערכות המבוססות על תפיסת ה-Grid לבין תפיסות שירותי הרשת וה-SOA בכלל, עתידות להביא אותנו למימוש תוכנה המבוססת על שירותים מבוזרים ומנוהלים.

3.3. עלייתם של פתרונות האינטגרציה

הרקע לעלייתם של פתרונות אלה נסקר במקורות רבים, כגון במחקר בהובלת ווליאם צאנג מאוניברסיטת ניו סוויט ויילס באוסטרליה [3]. עם זאת, ראוי לציין כי לנוכח ההשקעה הרבה של חברות ומוסדות אקדמיים מבחינת השינויים הרבים של המוצרים בתחום ולנוכח מורכבותו של התחום, אין זה מפתיע כי גם המוצרים הקיימים כיום אינם נותנים מענה הולם לחזון התוכנה המבוססת על השירותים. כיוון שהפורומים המקצועיים השונים טרם החליטו על פרוטוקול, אשר ייבטא מענה לצורך שהוצג והוא עוד רחוק מלהיות מוכן לכך, נדרשים, למעשה, עוד מחקרים רבים על מנת להגיע למענה כזה. אפשר לומר בבטחה כי רק דור המוצרים הבא או זה שלאחריו ייתן מענה הולם לתחום.

ראשיתו של גל משאבי המחשוב (Computing Utility), הניתן לצריכה בצורה דומה למשאבים ולשירותים האחרים, כדוגמת חשמל או טלפון, הופיע בספרות האקדמית עוד בשנות השישים של המאה ה-20. עם זאת, את יסודות מהפכת ה-Grid, שאנו חווים כיום, אפשר לשייך לפעילות המחקרית המאומצת בתחום החישוביות המקבילית והחישוב המבוזר, אשר התרחשה בסוף שנות ה-80 ובתחילת שנות ה-90.

פעילות מחקרית זו הבשילה לסדרת פתרונות תקשורת, חומרה, תשתיות התוכנה ושפות התוכנה, אשר אפשרו את בניית רשתות המחשבים, הכוללות קבוצות מחשבים שמריצות במקביל את חלקיו של יישום מקבילי בודד. במקביל, פתרונות התחלתיים של תשתיות

המחשוב המבוזר, כדוגמת CORBA, חשפו את הצורך (שלא סופק על-ידי פלטפורמות אלו) בפתרונות הסנכרון, בניהול התקשורת, בבקרת התהליכים, באבטחה ובסטנדרטים בין מערכות שונות, על מנת להגיע לחזון של שירותי מחשוב כמשאב [4]. עם זאת, פתרונות אלה באו לטובת ארכיטקטורה סגורה ותוך ארגונית. אפשר בהחלט להצביע על הדרישה לפתרונות אינטגרציה בין פלטפורמות כבעיה, שהיא בעיקרה אקדמית, עד לפריצת האינטרנט באמצע שנות התשעים. פריצה זו של האינטרנט בשילוב תקצוב מסיבי של התחום על-ידי גופי ממשל אמריקאיים, הביאה להקמת גופי תקינה, כדוגמת ה-Global Grid Forum, ב-2001 ולהבשלת פרויקטים ראשוניים המבוססים על ה-Grid, בשילוב עם תקנים ראשוניים. במקביל, הופעת האינטרנט גרמה להדיפת פתרונות אינטגרציה חדשים בין המערכות, כדוגמת מערכות ה-Enterprise Application Integration (EAI). עם זאת, אפשר להצביע כי פתרונות אלה כשלו, כיוון שהם היוו תרומה שולית מעבר לפתרונות העברת המסרים, אשר היו קיימים לפני כן בארגון. פתרונות ה-EAI היו רחוקים מלאפשר שימוש חוזר ברכיבי התוכנה, שלכל היותר היה בהם קירוב של החלקים השונים בארגון [4].

3.4. ה-SOA כתשתית אינטגרציה מלאה והופעת ה-ESB

אם פתרונות העבר כללו הבטחה גדולה, היום נראה כי עם הופעת מענה ה-Service Oriented Architecture אנו עומדים בפני מענה המאפשר מימוש הולם. תשתית שירותי הרשת (WS) מנצלת את ה-XML על מנת לספק ארכיטקטורה פתוחה וגמישה, אשר אינה מצריכה סימטריה בין שתי נקודות קצה ואינה תלויה בשכבות נמוכות יותר, כדוגמת תפיסות פיתוח, שפות תוכנה ותשתיות מערכתיות [27].

ה-SOA עושה שימוש בפתרונות ה-WS על מנת לחשוף שירותים באופן סטנדרטי בתוך הארגון ומחוצה לו. ארגונים רבים מבצעים חשיפה למערכות מורשת של היכולות באמצעות ה-SOA ומאפשרות בניית אפליקציות חדשות על בסיס שימוש ברכיבים הניתנים לשימוש חוזר על-ידי פיתוחם כשירותים. מימוש זה מאפשר צמצום של כפילויות בתוכנות שונות המממשות את הלוגיקה העסקית במערכות השונות [4].

החשיבות הגדולה של ה-SOA לתחום האינטגרציה היא יצירת ממשקים אחידים בכלל המערכות, דבר אשר מקל בצורה משמעותית על האינטגרציה ביניהם. בנוסף לכך, מיישם הגדרה משמעותית של כמה פונקציות שמספקת כל מערכת בצורה ישירה ועל כן מקלה על העברת מידע רלוונטי בין המערכות השונות [12].

הגידול בחשיפת השירותים ברשת מחייב פתרון תווכה, אשר יאפשר את גילוי השירותים בה, התמנות עליהם, אבטחת הגישה, ניהולה והעברת מסרים על גביה. זוהי תשתית ה-Enterprise Service Bus (ESB), אשר מהווה תקשורת שאינה תלויה בשפת התוכנה או במערכת ההפעלה ומאפשרת גישה לשירותי התשתית ולשירותים העסקיים בתוך גבולות הארגון ומחוצה לו [27]. הרחבה על נושא ה-ESB נמצא בנספח ב'.

על מנת לממש את פתרון ה-ESB הארגון נדרש לעמוד בשני יעדים [4]:

- הקמת סט שירותי תשתית, אשר יהיו זמינים דרך ה-ESB ויאפשרו סט שלם של אפליקציות חדשות. על הסט לכלול, קודם כול, שירותי תצוגה, ניהול שיחה, גישה לבסיסי הנתונים, ניהול האירועים וניהול תהליכים המבוססים על שירותי הניהול ברמת הביניים (עמידה במדיניות, כגון ביצוע Audit, ניהול טרנזקציות, ניהול זהויות וניהול התקשורת).
- סדרת כללי העיצוב והמלצות המימוש (Best Practices) לאנשי הפיתוח, אשר יאפשרו את הבטחת השמירה על הגמישות, על חוסר התלות בפלטפורמה (Java, .Net) ועל יכולת פעולה בין הפלטפורמות. במסגרת כללים אלה נכללים פרטוקולי ה-WS*, המתוקנים על-ידי גורמי תקינה שונים והנותנים מענה לניהול טרנזקציות, להחלפת מסרים מאובטחת, לניהול מדיניות האבטחה, להזדהות ועוד.

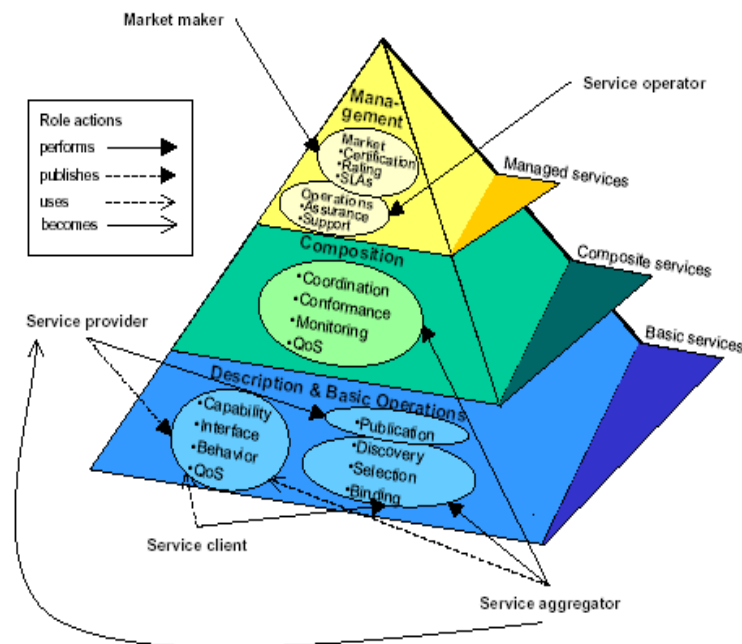
3.5. ה-ESB המבוסס על ה-Web Services וחסרונותיו

כפי שהוצג קודם, אחד המרכיבים הבסיסיים להקמת ESB הוא שימוש במסגרת של כללי פיתוח ועיצוב מוכרים. אוסף פרטוקולי ה-WS* בהחלט עונים על צורך זה, אך, כאמור, אין זה הצורך היחיד. בעוד ש-WS נותנים מענה לאבני הבניין הבסיסיות לפיתוח השירותים, הם אינם מספקים מענה לראייה הרחבה יותר. ראייה זו מבוססת על יצירת שירותים מורכבים ואפליקציות מתוחכמות, העושים שימוש באוסף רחב של שירותים בעלי מאפיינים שונים, המסופקים מספקים שונים והעושים שימוש באמצעים שונים. פירוט נרחב של חסרונות ויתרונות ה-Web Services בהקשרים אלה ראה [28].

לכן נשאלת השאלה, כיצד ספק שירות מורכב יכול להתחייב לרמת הזמינות, היתירות, האבטחה או אחר, אשר נתקף באמצעות חוזה השירות (SLA) בין ספק זה לבין הלקוח. זוהי סוגיה מורכבת עבור ספק השירות, העושה שימוש במשאבים המצויים תחת שליטתו. אך מה קורה כאשר הוא תלוי בקבוצת ספקי המשנה?

מחקרו של Papazoglou עוסק בסוגיה זו בהרחבה באמצעות הצגת פלטפורמת ה-Extended SOA (xSOA) [1], שמבוססת על שלוש שכבות שונות: האחת מבוססת על SOA ואפשר לזהותה כ-ESB (כפי שהוא מוכר לנו כיום). זוהי שכבת הביניים, המייצרת רכיבי ניהול עבור כל שירות והמאפשרת לו לספק את מדדי השירות בכל פרמטר רצוי ויכולת ניהול פרטנית, בין שהוא שירות פשוט או בין שהוא מורכב. השכבה העליונה מספקת את זירת המסחר והבקרה לאוסף השירותים ומאפשרת לארגון לשלוט על ה-SLA של השירותים המורכבים וניהול המסחר בהם על-פי המאפיינים. כאן ארוחב מעט יותר על כל אחד מן הרכיבים על מנת להבהיר את ההבדל בין ה-ESB לחזון ה-ESB המבוסס על ה-Grid.

1. שכבת בסיס המבוססת על ה-SOA – תפיסת ה-SOA היא אוסף של סטנדרטים ופרוטוקולים, המאפשרים להגיע למצב שבו ספק השירות יוכל לפרסם את שירותיו על-ידי הפצת תיאור ממשק השירות של מערכת התוכנה, שהוא מתפעל אותה. בעוד שצרכן השירות יוכל לגלות את השירות, לזהות את מאפייניו ולצרוך אותו. עם זאת, שכבה זו חסרה במיכון הטיפול במאפיינים השונים של השירותים השונים (איזה שירות להעדיף? האם מחיר השירות סביר? מה הזמינות שהוא מספק? איזו רמת אבטחה הוא מקיים? באיזה סוג הזדהות עדיף להשתמש? האם לחתום על המסרים?).
2. שכבת הביניים אחראית על ניהול איכות השירות ועל כך עונה (באופן הולם) על השאלות שאין להם תשובה:
 - א. תיאום וניהול התהליכים (תזמור אוסף השירותים עד להגעה לנקודת היעד).
 - ב. ניטור (יכולת התמנות על אירועי הניטור ויצירת דיווח ברמה הגבוהה יותר על-ידי איסוף האירועים, הסיכום או הסינון).
 - ג. עמידה תקינה (בדיקת תקינות הפרמטרים, עמידה בחוקה ובהיתוך המידע).
 - ד. איכות השירות (עלות כוללת במונחי זמן ומחיר, רמת האבטחה, האמינות, הזמינות ויתירות המידע והשירות).
 - ה. בקרת המדיניות (בקרה שהמחויבות כלפי הלקוח קוימה, כגון כלל המידע שנצרך עבר בהצפנה ברשת).
3. השכבה העליונה היא ניהול השירותים או ניהול זירת המסחר, שבה כל מערכת תוכל לבחור את סט השירותים הרצוי לה על-פי הצרכים העסקיים שלה, לדוגמה:
 - א. גוף הנדרש לפקח על אספקת השירות ברמת אמינות גבוהה מאוד וזמינות של חמש תשיעיות, כדוגמת מערכת לסליקת חיובי אשראי, יבצע רכש של שירותים העומדים בדרישות אלו ויאלץ להתפשר עם נותני השירותים על הנושא הכספי בעזרת משא ומתן.
 - ב. גוף הנדרש לספק שירות בסיסי וזול (Price Leader), כדוגמת משלוח דואר אלקטרוני, ילחץ על ספקי השירותים בנוגע למחיר, תוך התפשרות על זמני המשלוח (ההתחייבות על אורך התהליך מקצה לקצה).
 - ג. הגופים השונים יוכלו לבצע התאגדות לשם הגדלת נפח הרכש שלהם ולזכות בהנחת הכמות. שכבה זו של זירת המסחר תאפשר סחר בין ספקי השירותים (קבלני המשנה) לספקי השירותים המורכבים יותר או לספקי האפליקציות ללקוח הסופי ותכלול מענה לצרכי ניהול הזירה: הקמת שירות חדש, ניהול המדדים כדוגמת הספק וזמן התגובה, ניתוב דינמי לתמיכה בתרחישי כשל של השירותים ו/או איזון העומסים, ניהול מחזור החיים, כולל קישור למערכות הבקרה שיטפלו במקרי הכשל, ניהול קונפיגורציה בעת ריצה, ניהול גרסאות ותאימות לאחור ופתרון גילוי השירותים והרחבת הרשת.



איור 4 - ארכיטקטורת ה-xSOA [1]

Papazoglou מציג בעבודתו את שני התרחישים המרכזיים: ניהול זירת המסחר לשירותים וניהול שרשרת האספקה הלוגיסטית. בשני המקרים המיקוד הוא על נושא ה-SLA בין הגופים השונים, מתוך ראייה גלובלית של כל נותני השירותים מקצה לקצה, כולל המתחרים השונים ותוך שמירה על שקיפות מלאה בין הספקים השונים.

העמידה ב-SLA מתבססת על עמידה במדדים (Key Performance Indicators) הניתנים לכימות, ומשתלבים בתפיסת ה-IT כחלק מהתהליכים העסקיים של החברות.

בהסתכלות עמוקה יותר על ארכיטקטורת הפירמידה שהוצגה אפשר בהחלט להקביל את שכבת ניהול השירותים ואת שכבת ניהול זירת המסחר לטכנולוגיה קיימת, המתמקדת בדיוק בנושאים אלה – טכנולוגיית ה-Grid. טכנולוגיה זו התמקדה בהתחלה במתן מענה לחישוב המקבילי (הרצת אלגוריתם מסובך וכבד הדורש זמן ריצה ארוך על מספר רב של מעבדים במקביל על בסיס פירוק התהליך לסדרת תהליכים שניתן למקבל). בחישובים אלה יש לוודא את עמידתם של כל אחד מקצוות המערכת בזמן ריצה סביר, בהחזרת פלט תקין וזאת על מנת שהשלב הבאים לא יכשלו.

שילוב של טכנולוגיית ה-Grid, שקיימים בה ממשקים מבוססים, כדוגמת ה-OGSA עם טכנולוגיית ה-SOA, מאפשר יצירת פלטפורמה שלמה. זאת, כיוון ששתי טכנולוגיות אלו משלימות זו את זו: האחת עוסקת בגילוי ובשימוש בשירותים ללא תלות בפלטפורמה, בעוד שהשנייה עוסקת בניהול ובניצול יעיל של המשאבים. עקב הסתכלותנו על הפירמידה, סביר כי

תעלה השאלה, מדוע נדרשת זירת המסחר של השירותים בתוך הארגון, כאשר כל ספק של שירותי המידע הוא, למעשה, "מונופול" בתחומו? וזה בניגוד לשוק הבין-ארגוני, שבו שני גופי המסחר יכולים לספק שירותים דומים עם פרמטרי SLA שונים (לדוגמה חברי הבורסה המספקים שירותים מול המסלקה שלה). את התשובה לכך אפשר לחלק לשלוש:

א. גם בתוך הארגונים יש ספקי שירות בתימחורים שונים, לדוגמה שירותי שליחויות המנהל - כמה אפשרויות יש לקיום המשלוח? כגון אווירי או ימי. כל אחד מהם מסוגל למסור את החבילה בזמנים שונים, במהירויות שונות ובעלות אלטרנטיבית שונה. זירת מסחר יעילה תוכל להפנות כל אחד מאמצעי המשלוח האלה וכך להגיע לניצול מירבי של העלויות, המהירויות ושאר המרכיבים התחרותיים ("השוק החופשי").

ב. יישום אמצעים מעין אלה יאפשר את מימושם של מנגנונים הנוגעים לאיזון העומסים, התמודדות עם כשלים במתן השירות והעברה לעבודה מול שירות חלופי והכול בצורה גנרית ללא צורך בשימוש בהם. שימוש בפתרון זירת המסחר יאפשר לגוף ה-IT התוך ארגוני להקים מערכת עם גיבוי אקטיבי בעלות SLA גבוהה, אשר ינוצל רק עם נפילת המערכת המרכזית.

ג. מימוש מנגנוני זירת המסחר יאפשר לתמחר כראוי, עבור לקוחות תוך ארגוניים, את השירות, ועל-ידי כך לבסס את גוף ה-IT כמרכז רווח, אשר אינו תלוי בפרמטרי חיוב שירותיים, כגון מספר משתמשי הקצה בתת-היחידה המקבלת את השירות. סקירה מלאה של מיקום גוף ה-IT בארגון כמרכז רווח ואופן השירות של ה-SOA ניתן למצוא במחקר של מיכאל וורדר מאוניברסיטת ציריך [30], ועל משמעויות מודל התמחור לתחום ה-WS במאמר של גוף המחקר של ZapThink - [28].

מודלים אלה, למעשה, חושפים לנו את חסרונם הגדול של פתרונות ה-ESB הקיימים, הבנויים על תפיסה הרואה את השירותים כחברים וכנגרעים מ-ESB מרכזי, בעוד שבחזון העתידי נראית התממשקות דינמית בין תתי-ארגונים שונים המבוזרים באופן דומה. אפשר בהחלט לומר כי פתרונות התומכים בחבירת אד-הוק של שני ESB שונים טרם הופיעו ועתידים לתת מענה רק בדורות הבאים של מוצרי האינטגרציה.

3.6. הבנת הדרישות האוטונומיות של הפלטפורמה וביזור

הבעיות שהוצגו בארכיטקטורת ה-ESB, המבוססת על פלטפורמה מרכזית, גורמות לגופים השונים בעולם למצוא פתרונות. הגופים המרכזיים הנוגעים להתפתחויות של טכנולוגיות אלו הן הקהילות האקדמיות, הגופים התעשייתיים והממשלתיים.

הצורך לפתרונות מחשוב מבוזרים מהותי ומשמעותי בקהילה האקדמית עקב ריבוי המחקרים העוסקים במימוש סימולציות שונות (מזג האוויר, ניסויים גרעיניים), בבניית מודלים (מידול התפתחות השוק) ובעיבוד מידע רב (גילוי ישויות בחלל על בסיס תדרי רדיו) הדורשים שימוש בזמן עיבוד רב, אשר אינו בהכרח טורי, בצד התקציבים המקבילים.

בגופים הממשלתיים, ובעיקר הביטחוניים, נדרשת דינמיות רבה של הכוחות וחוסר תלות ביניהם לבין הגורמים האחרים, עקב החשש לחיי האדם. מצד אחד, דבר זה גורם לצורך

לאוטונומיות במשאבי המחשוב של הכוח, הכוללים יכולת התאוששות מהירה ללא כשל. מצד אחר, יש צורך במענה מהיר לחבירת כוחות מסוגים שונים ויכולת ראייה מלאה של שדה הקרב לכוחות בשטח ולדרג הפיקוד, בשל הדינמיות הרבה של שדה הקרב וההתרחשויות הלא צפויות. הצורך הנ"ל הולך ומתחזק עם כניסת מערכות המידע לשדה הקרב ואפשר לראותו בפרויקטים כדוגמת ה-Global Information Grid (GIG) של צבא ארה"ב. בסקירה שנערכה ב-Jane's [31] על הנושא הוצגה עבודתה של חברת Boeing, המיישמת במעבדות המחקר שלה את המענה למערכות האוטונומיות המסוגלות לבצע תיקון עצמי, בצד מענה לחבירת כוחות על בסיס תקשורתי.

דרישות אלו, המופיעות כיום בקרב הגופים האקדמיים והממשלתיים, צפויות לחדור גם למגזר העסקי, ככל שהארגונים יהיו תלויים יותר במערכות המחשוב ובמערכות השליטה והבקרה. הדבר דומה להתפתחות האינטרנט, שהחל במוסדות האקדמיים ובמימון ה-DARPA והגיע למה שהוא היום [4].

המענה ליכולת האוטונומיות והחבירה המהירה של המשאבים והכוחות המבוזרים מבוסס על מענה של וירטואליזציית המשאבים. אם כל ממשק פיזי מספק ממשקים לוגיים דומים, בין שהוא מרוחק או בין שהוא נמצא באתר, הרי שאפשר ליישם באופן פשוט יחסית מערכת אשר תיתן מענה דינמי לשלל המצבים [3].

פתרונות טכנולוגיית ה-Grid מתבססים על טכנולוגיות הווירטואליזציה ומתפתחים לכיוון של יצירת "שכבת וירטואליזציה" (Virtualization Layer) לאירוח שירותי הרשת (WS) [17].

3.7. ההתפתחות והפתרונות בעולם ה-Grid

לאחרונה, נראה שמערכות המבוססות על Grid יהיו פתרון מוביל לתחומי האינטגרציה בין מערכות ובין רשתות, ולמעשה, יובילו את תחום המחשוב אל עבר חזון ה-"Utility Computing", שבו שימוש בשירותים מבוססי תוכנה יהיה דומה לשימוש בחשמל בבית – זול, אמין ופשוט.

סקירה נרחבת של חזון זה והשלבים השונים של תחום ה-Grid מובאים בגיליון מיוחד של IEEE לנושא ה-Process Computing [6]. עם זאת, ראוי לציין את השלבים ואת הפתרונות המרכזיים שמומשו בתפיסה זו, על מנת להבין כי טכנולוגיה זו נמצאת בשלביה האחרונים לפני נקודת שיאה והכניסה לזרם המרכזי של טכנולוגיות המחשוב:

1. ניצנה הראשונים של תפיסת ה-Grid הופיעו בשנות השישים של המאה העשרים. מאז, משאבים ומאמצים נשגבים הופנו לטובת המחשוב המבוזר (בעיקר בשנות השמונים והתשעים). הם מהווים את אבני הבסיס של מערכות ה-Grid בדמות סדרה של מערכות התקשורת, החומרה, תשתיות התוכנה ושפות תכנות ייעודיות במוסדות המחקר. אפשר לזהות בתקופה זו ניצנים ראשונים של מוצרים מסחריים, כדוגמת ארכיטקטורת ה-CORBA.

2. פריצת הסכר בתחום ה-Grid התרחשה באמצע שנות התשעים עם פריצת האינטרנט, אשר חשף את היכולת לקשר כמויות גדולות של משאבי עיבוד מבוזרים – יכולת אשר

הייתה תאורטית עד נקודה זו. פריצה זו ארעה במקביל למימוש של כמה פרויקטים מחקרניים, כדוגמת ה-Globus Project, אשר מומנו בידי גופי הממשל האמריקאי, וייצרו תשתית המאפשרת לנצל את המשאבים המבוזרים הקיימים ברשת האינטרנט.

3. כינונו של ארגון מרכזי, העוסק בתחום ה-Grid, ה-Global Grid Forum, שנוסד ב-2001 על בסיס ארגונים קודמים מארה"ב, אירופה ויפן. הארגון מקדם את התקינה בתחום ארכיטקטורת ה-Grid, ומהווה גורם מרכזי למחקרים השונים.

בשנים האחרונות מומשו כמה פרויקטים מרכזיים, אשר הציגו את יכולות תפיסת ה-Grid ואת הערך המוסף הגבוה, אשר הוא מסוגל להניב לארגונים לא אקדמיים בעתיד. פרויקטים אלה מתוארים בהרחבה באסופת מאמרים שיצאה במסגרת ה-Proceedings of the IEEE בגיליון בשם Grid Computing [6]:

1. ה-Globus Project פיתח תשתית בשם Global Toolkit, הנותנת מענה לשכבת תווכה לאספקת השירותים המרכזיים של ה-Grid Service, שמתוארות בהמשך. פרויקט זה, שתוצרו האחרון הוא ה-GT 4.2.1, יצר את הסטנדרטים הראשונים של תחום ה-Grid.

2. פרויקט EGEE - Enabling Grids for E-science החל לפעול ב-2004 במימון של הנציבות האירופית. פרויקט EGEE מחבר שבעים מוסדות מחקר ועשרים ושבע מדינות של האיחוד האירופי, במטרה לבנות את תשתית ה-Grid כחלק תשתיות המחשוב של האיחוד האירופי עבור הגופים המדעיים.

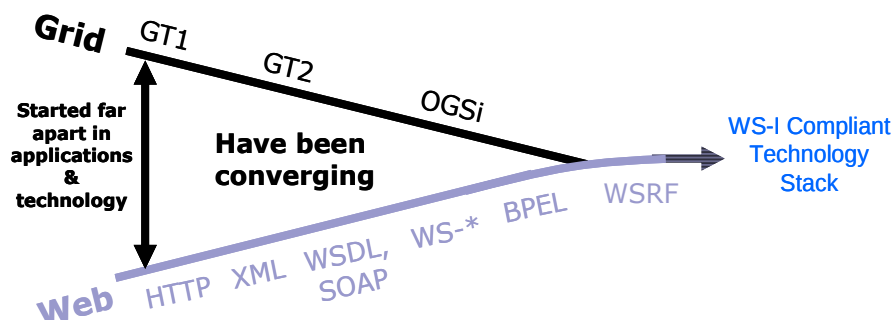
3. (ESG) "The Earth System Grid": מערכת המבוססת על ה-Grid ומיועדת למידול ולסימולציה אקלים כדור הארץ. הייחוד במימוש הטכנולוגי של המערכת היא ההכוונה לניהול, לגילוי, לגישה ולניתוח של מאגרי מידע מבוזרים גדולים מאוד, הכוללים מאפיינים רבים לכל רשומה. הפרויקט נוגע בהיבטים רבים, הקריטיים למימוש מערכת המבוססת על ה-Grid כדוגמת הזדהות, ניהול הרשאות וגישה, שינוע וניהול מידע רב, יכולת גישה מרוחקת למידע תוך עמידה בדרישות ביצועים מחמירות, ביזור המידע ושכפולו, קטלוג המידע וגילוי וניטור מבוזר של כלל המערכת.

4. DAME: פרויקט הנדסי למידול ויברציות ומערבולות, הנובעות מהפעלת מנוע סילוני מתוצרת Rolls-Royce. הפרויקט מיישם רשת נוירונים מתקדמת, על מנת לבצע חיפוש והתאמת התבניות בסדרות העיתיות (Time Series). ייחודו של פרויקט זה נעוץ ביכולת הביצוע של החיפוש וההתאמה על בסיס מאגרי מידע מרובי רשומות ועמודות, המצויים בכמה אתרים המבוזרים זה מזה.

5. "The Computational Chemistry Prototyping Environment": סביבה המממשת מענה לצרכים בתחום הכימיה החישובית והכימיה קוונטית, הדורשת, כידוע, התמודדות עם מספר רב של מצבים סטטיסטיים. סביבה זו, המבוססת על ה-Grid, מיוחדת בתמיכתה בניהול תהליכים מורכבים וגם בקישוריות בין בסיסי הנתונים המורכבים.

6. "Japanese Computational Grid Research Project" (NAREGI): פלטפורמה המבוססת על ה-Grid לצרכי מחקר בתחום הנאנוטכנולוגיה. זהו פרויקט לאומי המשלב מוסדות אקדמיים, ממשלתיים ותעשייתיים, והכולל מחקר ויישום של פלטפורמת תווכה המבוססת על ה-Grid בעלת יכולות ביצוע גבוהות, אשר אמורה לאפשר ליפן קפיצת מדרגה בתחום מחקר הנאנוטכנולוגיה.
- עם זאת, על-פי סקירתו של איאן פוסטר, חוקר מוביל בתחום ה-Grid [4], עדיין אין ברשותם של פתרונות המבוססים על ה-Grid להישען על מוצרי מדף (COTS), אשר תומכים בסטנדרטים מוגדרים וקבועים, אלא בעת פיתוחם. כיום, יש דרישה להסתמך על אוסף מוצרים, אשר נדרשת ביניהם אינטגרציה רבה. עם זאת, Foster מצביע על שתי מגמות מרכזיות, אשר תומכות בהתכנסות התחום בצורה דומה לזו שהאינטרנט עבר:
1. הסתמכות על תקנים: בדומה ל-IETF וה-W3C בתחום האינטרנט, קמו בתחום ה-Grid אוסף של גופים כדוגמת ה-Global Grid Forum (GGF), OASIS ו-DMTF המשקיעים מאמץ לתקנון שירותים וממשקים לניהול המשאבים, לניטור, לאיתות, להקצאה, לחילול ולחשבונאות. סטנדרטים אלה (WS-OGSA, WSRF, WS-Notification, WS-Distributed Management – Common Information Model) מאפשרים קישוריות ויכולת פעולה משותפת של מוצרים שונים מספקים שונים. בכך הם מקלים מאוד על קליטת המוצרים אצל הלקוחות. על כן, הם מהווים שלב בעל השפעה רבה לפני שהתחום יקבל תאוצה מסחרית.
 2. תוכנת קוד פתוח: ארגונים, השוקלים להיכנס לתחום זה כבר בשלב מוקדם נחשפים לסכנה בעת הסתמכותם על ספק יחיד, העושה שימוש בממשקים לא-סטנדרטיים, ולכן בעקבות הפלטפורמה הייעודית צפויה להם בעיה קשה של הגירה בהמשך. אחת הדרכים, אשר מאפשרות התגברות על חסם זה, שיושמה בהצלחה באינטרנט, היא שימוש בכלים מבוססי קוד פתוח (ראה לדוגמה Apache Web Server). כלים מעין אלה מונעים את התלות המידית בספק התוכנה, ולכן יאפשרו לבצע שינוי ממשקים, שינוי הקוד או ההגירה מן הממשקים הקיימים בקלות יחסית בעתיד. פתרון זה מאפשר הימנעות מהתקשרות עם ספק יחיד בשלב זה של מחזור החיים בטכנולוגיה.

כבר היום אפשר לראות כי בגופי התקינה מתבצע תהליך של קונסולידציה של תקנים בין תחום ה-WS לבין תחום ה-Grid.



איור 5 - איחוד סטנדרטי של ה-Web Services וה-Grid בדרך לתקן מרכזי WS-I

קונסולידציה זו תיתן מענה לשירותי רשת המבוססים על סדרת ממשקים קבועים וידועים הנותנים מענה לצרכי ה-Grid, וזאת בדרך דומה למחלקה היורשת ממחלקת אב בתכנות מונחה עצמים (OO). בין הממשקים המרכזיים אפשר למנות את אלה:

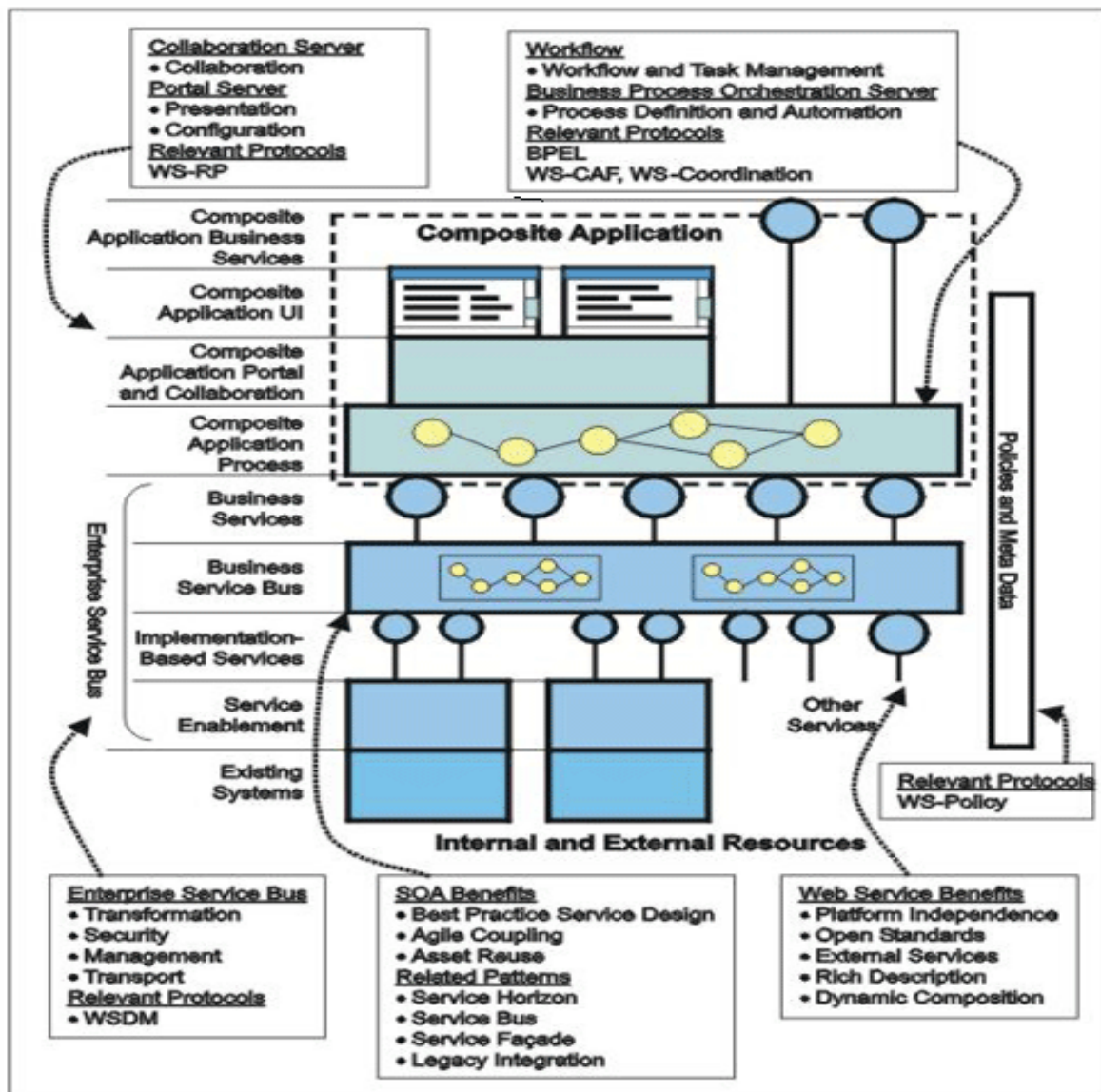
- גילוי (Discovery) – יכולת גילויים של השירותים הקיימים, הגדרת תכונותיהם ויכולת הגדרה עצמית.
- חילול דינמי (Dynamic Service Creation) – יצירת שירותי רשת חדשים בצורה דינמית, כולל שימוש במודלי Factory המוכרזים מתחום ה-OO, וניהולם.
- ניהול מחזור החיים (Lifetime Management) – יכולת יצירת השירות וניהול התלותיות בשירות.
- איתות (Notification) – יכולת לנהל שינויים מהותיים במקרים של כל סוגי השירותים ויכולת בקרה על מצב כלל השירותים מתוך ממשק מרכזי.
- ניהוליות (Manageability) – יכולת לנהל, למדוד ולבקר אוספים גדולים של שירותים.

אפשר למצוא מידע מעמיק יותר על כל אחד מן הנושאים הנמצאים בנספחים.

4. חזון ה-ESB

4.1. התפיסה

בפרק זה יוצג מודל הייחוס (Reference Model) לפיתוח יישומים מכווני שירותים תוך מימוש עקרונות ארכיטקטורת ה-SOA. כפי שאפשר לראות באיור 6, שכבת ה-ESB מהווה את אחת השכבות המרכזיות במודל זה. הבנת המודל הארכיטקטוני כולו, כולל השכבות השונות שבו, תפקידה של כל שכבה ואת ההקשרים ביניהן, הקריטיים להבנת המהות והתפקיד של שכבת ה-ESB במסגרת התמונה המלאה של ה-SOA.

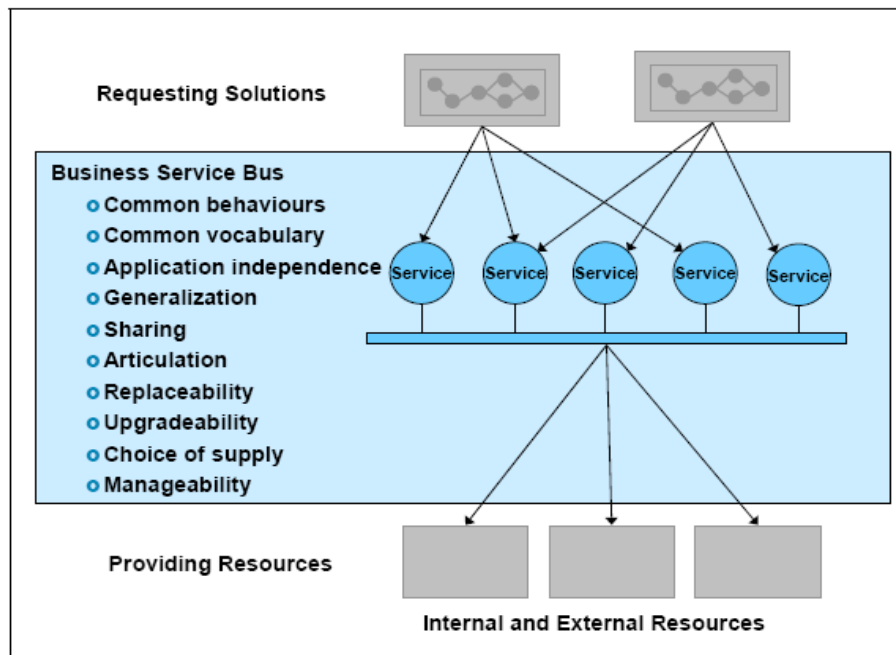


איור 6 - מבנה ייחוס שלם לארכיטקטורת ה-SOA [24]

4.2. שכבות מרכזיות בארכיטקטורת ה-SOA

להלן תיאור השכבות המרכזיות בארכיטקטורה:

- Composite Applications Layer (CA) – שכבה זו מכילה את אוסף היישומים/ פתרונות עבור משתמשי הקצה העסקיים של הארגון. היישומים נבנים במטרה לממש תהליכים עסקיים בעלי ערך ומשמעות למשתמש הקצה ולתמוך בהם. זוהי השכבה המרכזית (במושגים של ערך מוסף ועסקי למשתמש הקצה). שאר השכבות בארכיטקטורה אמורות לתמוך במימושה באופן ישיר או עקיף. שכבה זו שמה דגש על מימוש התהליכים העסקיים על-ידי ביצוע הקומפוזיציה (Composition), הקיבוץ (Aggregation) והתזמור (Orchestration) של שירותים שונים בשכבות התחתונות (בעיקר בשכבות ה-ESB ו-BSB). כפי שאפשר לראות באיור 6, תהליכים עסקיים, המוגדרים בשכבת ה-Composite application Business Services, יכולים להתגלות כשירותים עסקיים ברשת השירותים הארגונית. כיוון שהאוריינטציה של שכבה זו היא משתמש הקצה, השכבה מספקת גם תת-שכבה, האחראית על שכבת הממשק למשתמש הקצה (Composite Application UI). רכיבי שכבת ממשק המשתמש ירוצו בדרך כלל על גבי תשתית פורטל, אשר תספק את שירותי ה- Application Portal & Collaboration Services.
- Business Service Bus Layer (BSB) – שכבה זו מהווה תבנית ארכיטקטורה לוגית (Logical Architecture Pattern) לארגון קבוצת השירותים העסקיים למרחב עסקי מסוים (Business Domain). מטרתה המרכזית של תבנית זו לחלוק את סט השירותים העסקיים המוגדר ליותר מיישום/ תהליך/ פרויקט יחיד (Composite Applications) ולשתפו, כפי שאפשר לראות באיור 7. מימוש קונספט ה-BSB מאפשר ליצור תשתית מאורגנת ומסודרת של אבני לגו עסקיות (Business Services), על-פי תחומי העיסוק של הארגון. ה-BSB מבצע אגרגציה וקומפוזיציה של ה- Implementation Based Services עד לשיעור של שירותים עסקיים בעלי משמעות בקונטקסט של משתמש הקצה ושכבת ה-Composite Applications. מבחינת תהליכי הפיתוח של המערכות החדשות, משמעות הדבר היא שכל פרויקט חדש עושה שימוש באבני הבניין הבסיסיות הקיימות או, לחילופין, תורם להוספת אבן בניין חדשה לתשתית ה-ESB.



איור 7 - מבנה ה-Business Service Bus [24]

במסגרת מימוש ה-BSB נקבעת ונאכפת מדיניות (Policy) לסט של השירותים במרחב העסקי שמייצג ה-BSB. מדיניות זו מתייחסת להיבטי ה-Management (Investment, Semantics, Schemas, Rules, Design) ולהיבטי ה-Collaborations, Use & Reuse (Generalized Behaviors). אכיפת המדיניות בשכבת ה-BSB מבטיחה שיעדים עסקיים בסיסיים של העסק לא ייפגעו על-ידי יעדים נקודתיים, הרלוונטיים לפרויקט מסוים. הארכיטקטורה הלוגית של ה-BSB מספקת תשתית לאספקטים הבאים בהקשר לשירותים העסקיים המרכיבים את ה-BSB:

- התנהגויות משותפות לשירותים - Common Behaviors
- שפה משותפת לשירותים - Common Vocabulary
- חוסר תלות באפליקציה - Application Independence
- הכללת השירותים - Generalization
- שיתוף השירותים - Sharing
- גמישות - Articulation
- יכולת החלפת מימוש השירות - Replaceability
- יכולת שדרוג השירותים - Upgradeability
- יכולת בחירת ספק השירות - Choice of supply
- ניהוליות - Manageability

שכבת ה-BSB משקפת את הראייה המשותפת, אשר צרכני השירות וספקי השירות חולקים במרחב עסקי מסוים (Business Domain). לדוגמה, ה-BSB של יחידת ה-HR

בארגון יכול את סט השירותים הרלוונטיים ליחידה הארגונית HR וישקף את ההסכמה הקיימת בין הצרכנים של שירותי ה-HR בארגון וספקיהם. כבר כיום אפשר לראות בעסקים הוורטיקליים השונים, לדוגמה ביטוח, פיננסים, רפואה וכדומה את הופעתם של סטנדרטים (לדוגמה: ACCORD בביטוח, HL7 ברפואה). סטנדרטים אלה אמורים לאפשר את יצירת תמונת העולם המשותפת שצרכני השירות וספקי השירות חולקים בהקשר ל-Business Domain ספציפי דרך קונספט ה-BSB ולהקל עליה. שכבת ה-BSB מאפשרת את הצימוד הרופף (loose coupling) הנדרש בין היישומים בשכבת ה-CA לבין אוסף המשאבים הפנימיים והחיצוניים המספקים את המימוש לשירותים המוגדרים ב-BSB.

- Implementation Based Service Layer - שכבה זו, הנכללת במודל, אמורה לחשוף באופן שיטתי וסטנדרטי את המערכות/ המשאבים הקיימים. הפעילות בשכבה זו תתמקד ב-Web Service Enablement של מערכות ומשאבים קיימים (פנימיים וחיצוניים). בדרך כלל יהיה מדובר בעטיפה מבוססת סטנדרטים של WS לממשקים טכנולוגיים קיימים של המערכות השונות. הצרכנים הראשיים של השירותים בשכבה זו הם השירותים בשכבת ה-BSB.

- Enterprise Service Bus Layer (ESB) – שכבה זו מהווה תבנית ארכיטקטורית-לוגית המספקת מבט (Infrastructure View) אל יכולות התשתית הבסיסיות של הארגון, שכבת ה-ESB עושה זאת תוך הדגשה של שכבת התווכה בארגון (Middleware Layer). היא מספקת את היכולות הנדרשות למימוש תשתית הריצה והניהול, המשמשת את מגוון השירותים בארכיטקטורת SOA (גם שירותים עסקיים וגם שירותי תשתית). שירותי ה-ESB מהווים תשתית רוחבית המשמשים את כלל השכבות בארכיטקטורה. במימוש שכבת ה-ESB הושם דגש מרכזי ליכולת לשלב את יכולות התשתית הקיימות בארגון תחת תשתית אחת מרכזית.

4.3. הגדרת ה-ESB

למרות האמור לעיל, נכון להיום אין באקדמיה (ולא בתעשייה) תמימות דעים לגבי הגדרה ברורה וקוהרנטית של מושג ה-ESB או לגבי תיחומם של התכנים והיכולות תחת מושג זה. לדוגמה: **חלק** מן הספקים רואים בתשתיות תזמור השירותים (Service Orchestration) **חלק** מארכיטקטורת ה-ESB ואילו ספקים אחרים לא. יש גם ספקים אשר אורזים לתוך חבילת ה-ESB שלהם תשתיות/ מוצרי MOM ו-EAI. לכן אחת המטרות המרכזיות במחקר זה להגדיר את מהותו של מושג ה-ESB, לתחם את היכולות המרכיבות אותו, תוך התייחסות עניינית להקשר ולאימוצים שהארגון פועל בהם. רוב ההגדרות הקיימות ל-ESB מתייחסות ובאות מתוך עולם האינטגרציה ורואות ב-ESB סוג

של תשתית אינטגרטיבית ארגונית, אשר אמורה לתמוך בחיבורם של יישומים רבים ולאפשר להם לקיים שיחה, למרות ההבדלים ברמה הסמנטית והטכנולוגית ביניהם.

על-פי נקודת המבט של הארכיטקטורה הכוללת אפשר בהחלט לראות ב-ESB יכולת (או יותר נכון, אוסף של יכולות), אשר מאפשרת (Facilitates) את מימוש שכבת ה-BSB ושכבת ה-Composite applications. אפשר להסתכל על ה-ESB משתי בחינות שונות:

- **מבחינת הארכיטקטורה** ה-ESB הוא אוסף של ISB (Infrastructure Service Bus) סטנדרטים ופרוטוקולים מוסכמים המאפשרים את מימוש שכבת ה-BSB ושכבת ה-Composite applications במודל. ארכיטקטורת ה-ESB מהווה יישום של SOA בשכבת ה-Infrastructure Services.
- **מבחינת התפיסה של מוצר (או שילוב של כמה מוצרים)** אפשר להתייחס ל-ESB במונחים של תוכנה, מוצר(ים), אשר מאפשרים את מימוש שכבת ה-ESB בקונטקסט של SOA.

הארגון הראשון (מאז 1999), שהתמודד עם הגדרות הפלטפורמה החדשה, כולל את ההגדרה של השירותים התשתיתיים, כחלק מהעבודה על ה-Grid המידע העולמי, היה משרד ההגנה האמריקאי (GIG – Global Information Grid). לפי תפיסת צבא ארה"ב [22], ה-GIG מוגדר כ"אוסף שירותי מידע, תהליכים, ואנשים המקושרים ברשת גלובאלית לשם איסוף, עיבוד, אחסון, ביזור וניהול מידע לפי דרישה עבור לוחמים, מקבלי החלטות, וגורמי תמיכה. ה-GIG כולל את כלל תשתיות התקשורת, תשתיות מחשוב, תוכנה, שירותים, מידע מערכת, שירותי אבטחה ושאר יכולות קשורות הנדרשות להשגת עליונות מערכתית עבור צבא ארה"ב".

ארגון DISA (Defense Information Systems Agency) סיווג את ה-GIG כ"Grid מידע יחידי ומאובטח המספק מגוון יכולות בצורה שקופה" והגדיר ארכיטקטורה תומכת המורכבת משירותי תשתית ארגוניים הנקראים [22] GIG Enterprise Services (ES). ה-ES GIS מאחד אוסף של שירותי תשתית, המספקים את הבסיס עבור מרכיבי GIG נמוכים יותר עד לחייל הבודד. לאחר השלמתו של ה-GIG, הוא יקשר את כלל הטכנולוגיות של משרד ההגנה האמריקאי, בדומה לרשת האינטרנט.

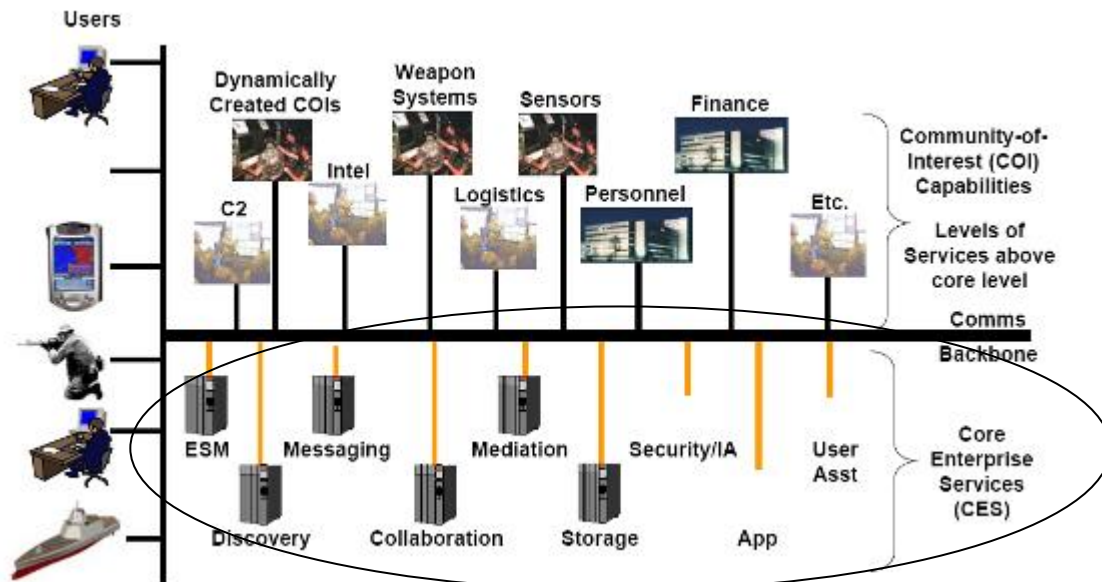
4.4. מרכיבי ה-ESB

לצורך בחינת מרכיביו הנדרשים של יכולות ה-ESB, אתבסס על עבודת הניתוח אשר נעשתה על-ידי משרד ההגנה האמריקאי ו-DISA במסגרת תכנית GIG, ובפרט על תת-התכנית NCES (Network Centric Enterprise services). בתת-תכנית זו מופו שירותי התשתית המסופקים על-ידי סביבת המחשוב העתידית לתשע משפחות שירותים, המרכיבות את תכנית ה-NCES כבסיס ל-ESB Service Model הארגוני [32]. על-פי תפיסה זו, תכנית ה-NCES מגדירה את תשע המשפחות הבאות של שירותי התשתית ברשת או תשע המשפחות של השירותים הבסיסיים:

1. שירותי אבטחת מידע (Security Services) – תמיכה בניהול הגישה (ניהול זהויות, Authentication, Authorization, אכיפת הגבלות הגישה), ניטור בטחון מערכות מידע, הגבלת הגישה בין תחומים (domains). שירותים אלה יופעלו על-ידי ספקי השירותים וצרכיהם על מנת לעמוד במדיניות האבטחה של הארגון.
2. שירותי שינוע המסרים (Messaging Services) – העברת המסרים בין האפליקציות, בין המשתמשים, תמיכה ב-Publish/ Subscribe, מסרים מידיים. שירותים אלה משלימים את שירותי השיתוף, התיווך והגילוי וגם מאפשרים לגשת למידע מכל מקום ובכל זמן.
3. שירותי אירוח (Application Services) – שירותים הניתנים לאפליקציה על מנת שיהיה אפשר לנהל ולשלוט עליה, לקבל עמידות ו-Scalability של האפליקציות. שירותים המאפשרים את ביצוע פריסתן של האפליקציות ובדיקתן.
4. שירותי ממשק המשתמש (User Assistance Services) – תמיכה ב-Agents המעבירים רק את המידע הדרוש למשתמש מכל רחבי הרשת, בשיתוף האנשים בתהליכים הארגוניים (Human Workflow) ובהצגת המידע ללקוחות השונים (סוגי תצוגה שונים).
5. שירותי אחסון המידע (Storage Services) – שירותים הנותנים יכולות לשליפה מהירה של הנתונים בארגון, תוך התחשבות במגבלות אבטחת המידע. נוסף לכך, שירותים אלה יתמכו בניהול אחסון המידע בארגון (גם באופן לוגי וגם באופן פיזי) ויאפשרו גישה למידע רב, אשר יאוחסן במקומות שונים ברחבי הרשת.
6. שירותי שליטה ובקרה (Management Services) – שירותי ניהול, ניטור ובקרה של השירותים/התשתית בארגון, המאפשרים גם לנהל את ה-SLA Service Level Agreement של השירותים בארגון ואת מחזור החיים של המשאבים. וידוא כי שאר השירותים זמינים ונגישים למשתמשים ושיש עמידה ב-SLA שהוגדר עבורם.
7. שירותי הגילוי (Discovery Services) – שירותים התומכים בגילוי השירותים, המשאבים והמידע ברשת, יצטרכו לענות גם על צרכי האפליקציות המשתמשות בהם וגם על צרכי המשתמשים בהם. בסביבה, שנוספות אליה מערכות חדשות ומערכות אחרות מוחלפות, יש צורך בשירותי גילוי, אשר ייחברו בין ספק השירות לבין הצורך בשירות.
8. שירותי שיתוף (Collaboration Services) – שירותים המאפשרים לשתף מידע בין תחומים (Domains) שונים בארגון, יאפשרו למשתמשים לעבוד ביחד ולבצע משימות משותפות. הם משלימים את שירותי השיתוף, התיווך והגילוי וגם מאפשרים להציג את המידע מכל מקום, בכל זמן ועל כל מכשיר קצה.
9. שירותי תיווך (Mediation Services) – תמיכה בתהליכים בין שירותים ובשירותי טרנספורמציה של המידע, מהווים את הדבק בין השירותים האחרים. השירותים יאפשרו, בין השאר, לבצע אינטגרציה והיתוך המידע, לעבד ול"תרגם" את המידע העובר בין ספק השירות לבין צרכן השירות.

בעזרת שירותים אלה אפשר יהיה לבנות, להפעיל ולנהל שירותים ותהליכים "עסקיים" מורכבים יותר וגם אפליקציות מורכבות למשתמשי הקצה. יש לשים לב כי השירותים אינם ניתנים רק באמצעות התוכנה, אלא גם באמצעות גופים, אשר מספקים שירותים בסיסיים, כגון: אירוח השירותים, אספקת שרתי האחסון, תמיכה למשתמשים בארגון ועוד. הדגש במסמך זה הוא על שירותים הניתנים על-ידי מערכות התוכנה כתשתית אפליקטיבית לשירותים האחרים.

מכאן שתפיסת ה-ESB מתרכזת באיגוד ובארגון של יכולות התשתית בארגון (היכולות אשר תומכות בשכבת היישום). תפיסה זו גורמת לשינוי מודל העבודה בארגון, והפיכה הדרגתית של הארגון לספק השירותים (Service Provider). המעניק ללקוחותיו שירותי תשתית במודל של "שירותים על-פי דרישה" Services On Demand, תוך ניהול ואכיפה של הסכמי השירות ורמות השירות, אשר הוגדרו בין צרכני השירותים וספקיהם. איור 8 מתאר את מיקום ה-ESB בחזון ארכיטקטורת שירותי הרשת של משרד ההגנה האמריקאי ומספק תיאור ויזואלי של הקשר, שיש לראות בו את תשתית ה-ESB.



[22] איור 8 - שירותי התשתית של ה-GIG

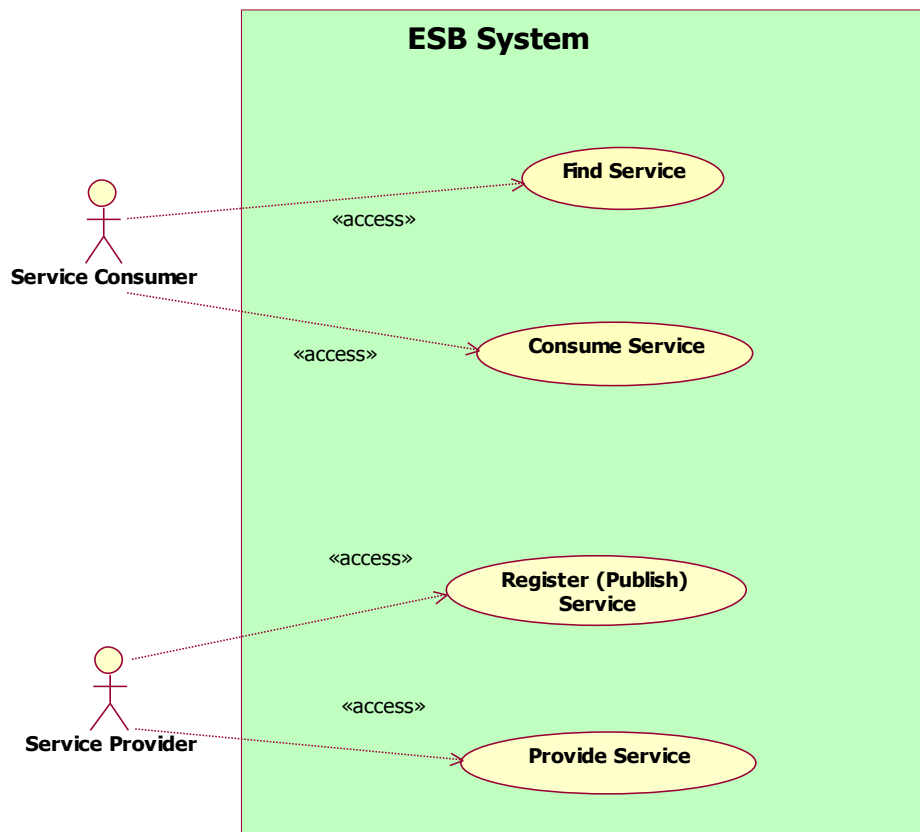
באופן כללי, אפשר לומר כי, ה-ESB מהווה גורם מאפשר מרכזי (Enabler) למימוש ארכיטקטורת ה-SOA בארגון, אם ה-SOA הוא הרעיון או הארכיטקטורה, אזי ה-ESB הוא אוסף יכולות התשתית המאפשרות לממש את הרעיונות הגלומים ב-SOA. יש לציין כי הרעיונות של SOA נוגעים לכך שה-ESB עוזר לממש מיושמים באופן פנימי בתשתיתו.

4.5. תסריטי שימוש ל-ESB

תסריטי השימוש המתוארים באיור 9 מתארים את האינטרקציות הבסיסיות ביותר הנמצאות ברמה גבוהה, שמבצעים שני המשתמשים העיקריים של תשתית ה-ESB בארגון:

- צרכן השירות - Service Consumer
- ספק השירות - Service Provider

ESB - Basic Use Cases



איור 9 - תסריט שימוש בסיסי

ברמת-העל המתוארת אפשר להגיד שהתכלית הבסיסית של תשתית ה-ESB, השייכת לארגון, לספק את התשתית הנדרשת לביצוע האינטרקציה בין צרכני השירותים לספקיהם בארגון. על התשתית לספק כלים מתאימים למימוש אינטרקציות אלו בצורה מאובטחת (Security) ומנוהלת (Management) תוך אספקת רמת השירות המובטחת (QoS). התשתית אמורה לספק את היכולות לגשר על פני פערים טכנולוגיים וסמנטיים הקיימים בין צרכני השירותים לספקי השירות.

פרספקטיבת צרכן השירות (Service Consumer)

הפעילויות הבסיסיות שצרכן השירות (Service Consumer) מבצע מול ה-ESB מנקודת מבטו:

- איתור/ חיפוש השירות (Find Service)
- צריכת השירות (Consume Service)

פרספקטיבת צרכן/ ספק השירות (Service Provider)

הפעילויות הבסיסיות שספק השירות (Service Provider) מבצע מול ה-ESB מנקודת מבטו :

- **רישום השירות (Register Service on ESB)**
- **אספקת השירות (Provide Service)**
-

4.6. המאפיינים העיקריים של ESB ארגוני

אפשר לקטלג מאפיינים אלה לשמונה מאפיינים בסיסיים לתשתית ESB ארגונית. על-פי התפיסה המתוארת במסמך זה, הזכות להיות מוגדר כתשתית ESB ארגונית עומדת ביחס ישיר למידת העמידה בהם :

1. אוניברסליות (Universality)

תשתית ESB אמורה לספק שכבת קישוריות אוניברסלית המאפשרת והמרחיבה את יכולות האינטגרציה מעבר לגבולות הארגון. על תשתית ה-ESB להיות מבוזרת ולתמוך במודלים השונים של Deployment וביכולת לבזר את סט היכולות שלה בצורה גמישה ברשת הארגונית.

2. הטרוגניות (Heterogeneity)

תשתית ESB אמורה לספק את יכולות ה-Middleware, האינטגרטיביות להפצה, לשינוע ולעיבוד מסרים בין מגוון רחב של מערכות ויישומים המשקפים סביבה טכנולוגית, הטרוגנית ובעלת רמת שונות גבוהה.

3. אינטראופרבייליות (Interoperability)

על תשתית ESB לעשות שימוש בפרוטוקולים פתוחים (Open Protocols), על מנת לתמוך באינטראופרבייליות בין תשתיות middleware של ספקים שונים המשולבים בארכיטקטורת ה-ESB. על תשתית ה-ESB לממש את הסטנדרטים של ה-WS-*, (SOAP), (WS-Policy), (WS-Security), (UDDI), (WSDL) כתנאי בסיסי.

4. אינטגרציה מתמשכת (Incremental Integration)

תשתית ESB אמורה לאפשר פריסה אינקרמנטלית של יכולות האינטגרציה הבסיסיות, על-פי הצורך והתקציב של הפרויקט/ ארגון. אפשר להתחיל ביישום בסיסי (ובעלויות נמוכות) ואז לגדול ולהתרחב על-פי הצרכים.

5. רמת השירות (Qualities of Service)

תשתית ESB אמורה להיות מסוגלת להגדיר ולאכוף את רמת השירות בנושאים הבאים : רמת הביצועים (Performance), זמינות (Availability), טרניקציות (Transactions), אבטחת המידע (Security), אמינות (Reliability) ושמירת המצב (Persistence).

6. תחלופתיות (Substitution)

על תשתית ESB להשתמש ב-Open APIs על מנת לאפשר ולתמוך ביכולת להחליף את מימושי התוכה (MW) בין ספקי התוכה השונים.

7. מכוון האירועים (Event Orientation)

תשתית ESB אמורה לספק תמיכה מובנית בקונספט של "אירוע" (Event), תוך כדי מתן יכולת להקטין את הצימוד בין יישומים המפרסמים (Publish) אירוע עסקי (Business Event) ליישומים המתמנים (Subscribe) עליו.

8. מכוון שירותים (Service Orientation)

תשתית ESB אמורה לספק תמיכה מובנית בקונספט של "שירות" (Service) תוך כדי תמיכה בצימוד רופף בין רכיבי תוכנה שונים.

4.7. מגבלות ה-ESB במצב הקיים

כיום, יש כמה גורמים המונעים מתשתיות ESB הקיימות להפוך לתשתית הבסיסית למימוש הפרדיגמה של סביבת מחשוב על-פי דרישה (On Demand Information Environment) ובאופן מבוזר:

- מרבית הפתרונות הקיימים מבוססים על מתווך מרכזי (ברוקר או תשתית BUS מרכזית). ישנן מספר בעיות המקשות את ההסתמכות עליו:
 - יכולת גידול מוגבלת (Scalability)
 - נקודת כשל יחידה (Single Point of Failure)
 - רמת ביצועים מוגבלת עקב המעבר דרכו (Performance)
- רוב פתרונות ה-ESB המצויים כיום מהווים אבולוציה טבעית של פתרונות ה-Application Integration ומתמקדים במתן פתרון לבעיית האינטגרציה בין היישומים השונים. יש צורך בתשתית אשר תאפשר מתן שירותים במודל של שירותים על-פי דרישה (On Demand Information Environment).

על מנת להגשים את חזון השירותים ברשת, על סביבת המחשוב הארגונית לעבור טרנספורמציה לכיוון של סביבה מונחית שירותים, המספקת את שירותי הארגון באופן דינמי ועל-פי דרישה. הרשת אמורה לאפשר לנו לחבר כל דבר (משאב) לכל דבר (משאב) במודל סטנדרטי ואחיד, תוך כדי קיום תמיכה מלאה בהעברת המסרים בין כל נקודת קצה אחת לכל נקודת קצה אחרת ברשת. כל האינטרקציות הללו אמורות להתבצע באופן מבוזר, מאובטח ומנוהל לחלוטין על-ידי תשתיות מתאימות שהרשת מספקת [17]. הרשת תתמוך ותאפשר לבצע בצורה סטנדרטית את הפעולות הבאות – הגדרה לוגית וניהול המשאבים המחוברים, חשיפה (Exposure), שיתוף (Sharing), צריכת (Consuming) כלל היכולות הפנימיות של המשאבים ברשת (גם היישומים וגם תשתיות מחשוב אחרות) להתבצע באופן מאובטח ומנוהל לחלוטין. על תשתית כזו להיות בעלת כמה מאפיינים (שאינן אפשרות למצוא את מימושם בפתרונות הקיימים כיום [24,26]) כפי

פונקציות לדוגמה	יכולת
Self Healing Self Configuration Self Optimization Self Protecting	מחשוב אוטונומי Autonomic Computing
Provisioning Deployment Configuration Reservation	שירותי ניהול המשאבים Resource Management Services
Job Management Execution Planning Workload Management Workflow Management Application Management	שירותי ניהול הריצה Execution Management Services
Heterogeneity Management Optimization Service Level Attainment QoS Management	שירותי ניהול השירותים Self Management Services

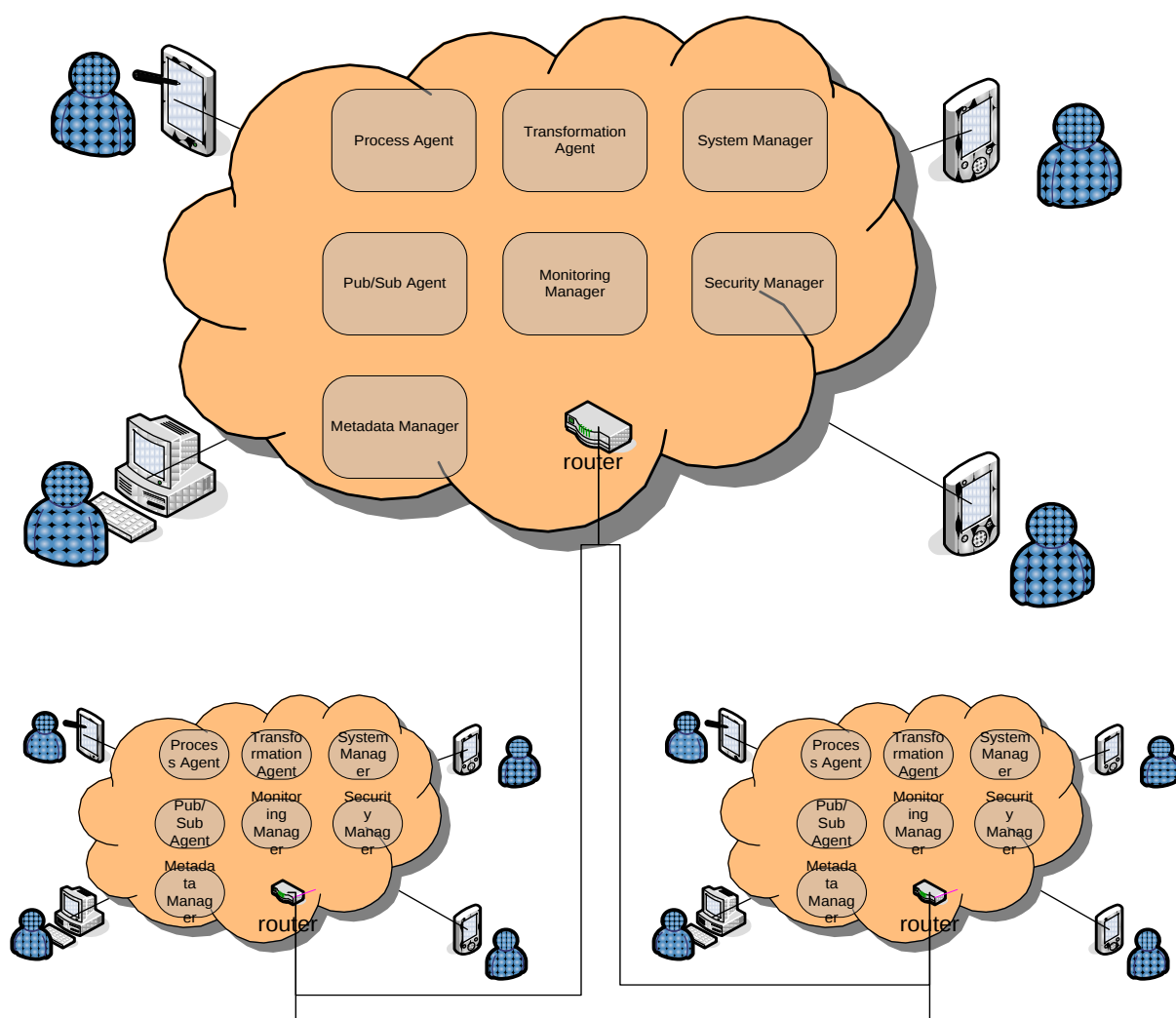
טבלה 1 - מאפייני התשתית On Demand Information Environment

4.8. ה-ESB המבוסס על ה-Grid

ננסה למפות את הכיוון שתפיסת הרשת ה-SOA בעידן ה-Grid מכוונת אליו מתוך הסקירה התיאורטית.

התפתחות תחום המחשוב ה-Grid ברמה האפליקטיבית בשנים האחרונות הביאה ליצירת סטנדרט חדש בתחום [10,15]. ה-Grid מורכב מאוסף רב של משאבים הטרוגניים, אשר מבצעים אינטרקציות ומנוהלים תחת מודל סטנדרטי מוגדר ועקבי.

ההשערה הבסיסית שהמחקר מכוון אליו הוא שאפשר ליישם את ה-ESB כחלק מפתרון המבוסס על-פי פרדיגמת ה-SOA באמצעות ארכיטקטורת OGSA, כך שעל תפיסת תשתית השירותים והאינטגרציה הארגונית, כפי שהיא מוכרת כיום, להפוך לתפיסה של רשת שירותים המבוזרת בתשתיתה. רשת זו תאפשר את חשיפת השירותים באופן סטנדרטי על גביה ואת צריכתם, תוך אפשרות להעברת מסרים מכל נקודת קצה לכל נקודת קצה (על-פי ההרשאות) ללא צורך במתווך מרכזי ובאופן מנוהל ומבוזר לחלוטין, כפי שמתואר באיור 10.



איור 10 - הדגמה ויזואלית לתפיסת רשת השירותים [24]

רשת האינטגרציה המושלמת תהיה מסוגלת לעמוד בדרישות הביצועים, ביכולת הגידול, ובשרידות המשמעותיים ביותר [2,29]. תפיסת הפתרון נשענת בברור על רעיונות מכמה תחומים חשובים בעולם המחשוב הנוכחי, כגון SOA, Autonomous Computing ו-Grid Computing. דוגמה לחשיבותה של תפיסה מורכבת זו אפשר למצוא בתכנית המחקר של משרד ההגנה האמריקאי, המשקיע רבות בבחינת תשתית כזו ובפיתוחה במסגרת תכנית ה-NCES - Network Centric Enterprise Services, אשר מנסה לבחון את הארכיטקטורה הנכונה למימוש פרדיגמה זו, כתשתית מבוזרת לפיתוח אפליקציות ומערכות לחימה של מערך ההגנה בארה"ב [7,22¹].

¹ ראה נספח ה' – דוגמה לפעילות מחקרית בנושא ה-SOA וה-Grid.

5. ארכיטקטורת ה-OGSA

5.1. הרעיון המסדר

נתאר תסריט של ארגון, אשר כלל המערכות (Systems) הפועלות בו אופיינו, עוצבו ופותחו על בסיס ארכיטקטורה וסטנדרט אחידים. כלל התהליכים הקשורים למחזור החיים של מערכות חדשות מתבצעים על בסיס של תקן ארכיטקטורה אחיד וסטנדרטי. התקן המספק מענה למגוון הקריטריונים שאליהם יש להתייחס במהלך גיבוש הארכיטקטורה, כגון: זמינות, שרידות, יתירות, אבטחת המידע, יכולת ההתרחבות, יכולת פעולה הדדית, ביצועים, אבטחת האיכות (QoS), ניהול ושיתוף המשאבים ועוד. תקן ארכיטקטורה סטנדרטי כזה יכול לשמש למגוון רחב של מערכות, בין שמדובר במערכות תשתית או, לחילופין, בין שמדובר במערכות ייעודיות בתחומי עיסוק שונים.

בעולם כזה, כל מערכת (System) נבנית מן היסוד על בסיס ההנחה כי כל יישום המשולב בה או במערכת אחרת הם חלק מהתמונה הכללית. כל מערכת או יישום מהווים אוסף של יכולות – טכניות ועיסקיות כאחד, אשר חוברות זו לזו כ-Grid שירותים כולל, ובכך יוצרות סביבת מחשוב אינטגרטיבית עבור כל צרכן אפשרי שלהן, וכמובן גם למשתמשי הקצה. אוסף השירותים הזה מאפשר לנו לבנות אפליקציות באופן גמיש, תוך הסתמכות על העובדה שהן עונות לתקן אחיד ומדברות באותה שפה.

תסריט כזה מעלה שאלות רבות ובראשם:

- האם התסריט המתואר כאן נשמע דמיוני?
- האם בכלל נעשים היום ניסיונות בידי האקדמיה או התעשייה להתאחד סביב תקן שכזה?
- האם ייתכן כי כבר היום אפשר להצביע ולראות איך תעשיית התוכנה (כחלק מתהליכי ההתבגרות והבשלות שלה) עושה צעדים ראשונים ומשמעותיים במסע ליצירתו של תקן אחיד לארכיטקטורה לפיתוח המערכות?
- האם בכלל ייתכן תקן ארכיטקטורה אשר יינסה להתמודד עם המחסור בסטנדרטים ארכיטקטוניים אחידים בעולם הפיתוח של מערכות ויישומים? תקן אשר יביא סוף לצורך הלא-הגיוני להגדיר את הארכיטקטורה למערכת כל פעם מחדש?

ובכן, התשובה (לפחות לחלק מהשאלות) היא שיש תקן ארכיטקטורי כזה, אשר נקרא OGSA (Open Grid Service Architecture). זהו הניסיון הראשון של תעשיית התוכנה (בהובלת קהיליית הקוד הפתוח GGF ובגיבוי התעשייה וגם בהובלת החברות התעשייתיות) לתת מענה לנקודות שצוינו.

המטרה העיקרית של פרק זה להתרכז בתיאור תקן ארכיטקטורת ה- OGSA תוך מתן הכוונה נכונה לתקן שלו מול מגוון הפעילויות והנושאים המטופלים כיום באקדמיה ובקהילת הקוד הפתוח.

5.1.1. תקן OGSA והקשר לעולם ה-SOA

המטרה המרכזית של תפיסת ה-SOA היא לאפשר בנייה של מערכת חדשה, המספקת בסיס לרשת אינטגרציה מוכוונת שירותים, עבור הקישוריות הבין-ארגונית והתוך-ארגונית. לפיכך אפשר לראות את תפיסת ה-SOA כמתמקדת בבדיקת ההתכנות לבניית הדור הבא של תשתית האינטגרציה, תוך כדי מימוש תקן ארכיטקטורת OGSA ויתרונותיה, ככלי ליצירת תשתית האינטגרציה.

5.1.2. תקן OGSA - רקע כללי

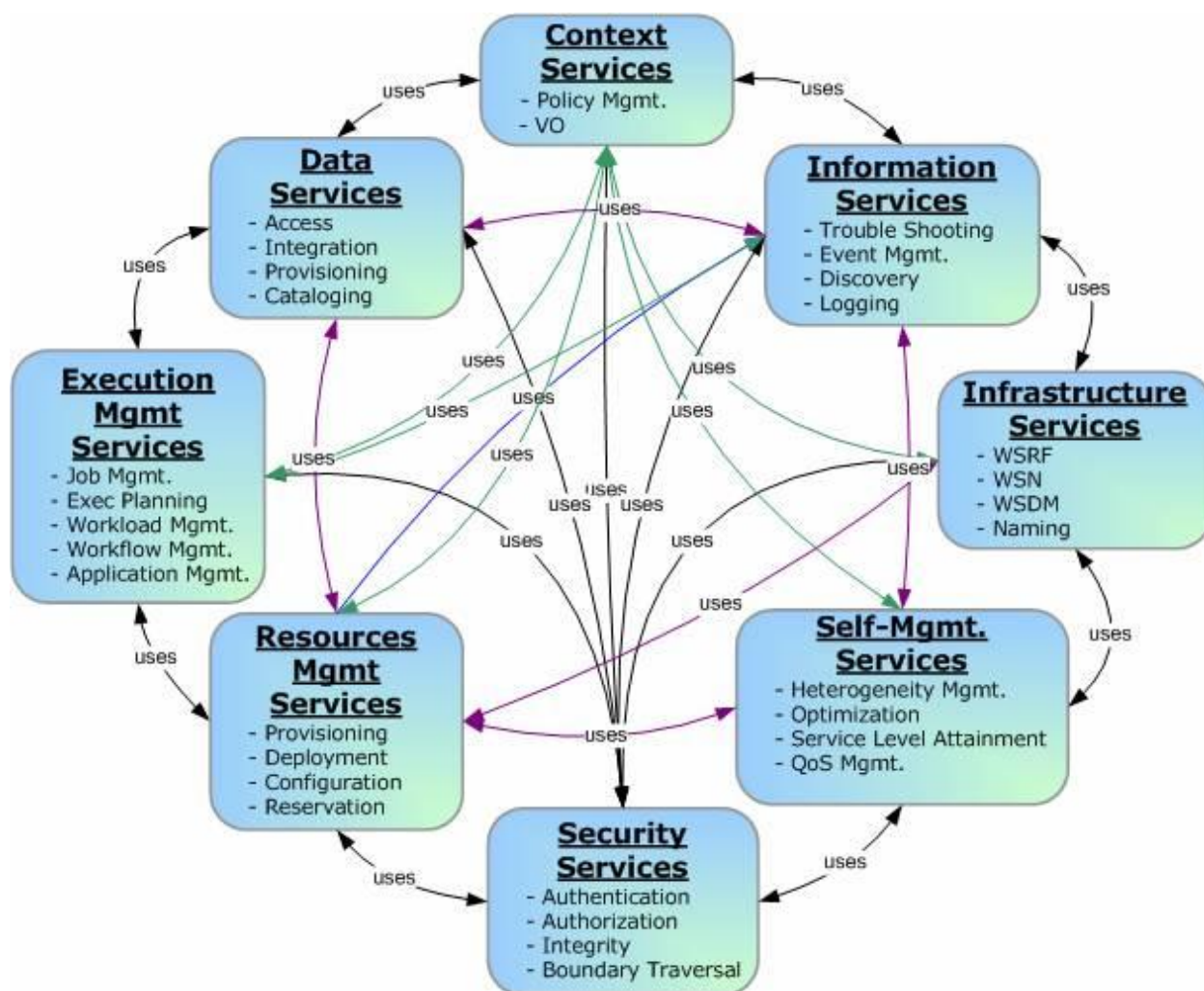
במרכזו של כל יישום או מערכת, המפותחת בתצורת ה-Grid, עומדת השאיפה למימוש הקישוריות, הווירטואליזציה, ניהול המשאבים והשירותים בסביבה המבוזרת, ההטרוגנית והדינמית. השגת מטרה זו מצריכה לפרוץ חסמים טכנולוגיים ולא טכנולוגיים רבים, אשר בדרך כלל מפרידים ומונעים את הסינרגיה הטבעית בין פלטפורמות מחשוב ויישומים שונים בתוך הארגון או בין הארגונים השונים. הצלחה בהפלת חומות אלו משמעותה שאפליקציות, שירותים, מידע ושאר המשאבים בתוך או מחוץ לארגון יהיו נגישים, בלי קשר למיקומם הפיזי. המפתח המרכזי להגשמת חזון ה-Grid הוא סטנדרטיזציה, כך שמגוון הרכיבים השונים, אשר מרכיבים את מארג המחשוב המודרני יהיו נגישים, אפשר יהיה להקצות אותם, לנטר את פעילותם, לאתר אותם ולחייב בגינם. כל זה יתרחש תוך כדי שהם מנוהלים כמערכת וירטואלית אחידה, אפילו שהיא נבנתה בידי כמה ספקים ומנוהלת בידי כמה ארגונים שונים. כאשר ברצוננו לממש מערכות ורכיבים בעלי אלמנטים של אינטגרטיביות, ניווד בין פלטפורמות ושימוש חוזר, סטנדרטים וסטנדרטיזציה הם גורמים בעלי משמעות. כמו כן, הקפדה על סטנדרטיזציה תורמת לפיתוח מערכות המבוססות על תצורת ה-Grid, אשר ניחנו בתכונות של עמידות, שרידות ויכולת גידול, אשר מאובטחות תוך הקפדה על שימוש ב-Best Practices. בפרק זה אציג ארכיטקטורה עתידית המבוססת על שירותים (SOA), במודל ובסטנדרט הנקרא Open Grid Service Architecture (OGSA), אשר באה לענות על הצורך בסטנדרטיזציה באמצעות הגדרת סט היכולות המרכזיות וההתנהגויות, אשר נותנות מענה על הבעיות העיקריות ביישום המפותח בתצורת מערכות ה-Grid (Grid Application). הבעיות הללו כוללות שאלות, כגון:

- כיצד מקבלים זהות וכיצד מתבצע אימות הזהות במערכת?
- כיצד מתבצע משא ומתן לגבי המדיניות וכיצד היא באה לידי ביטוי?
- איך מאתרים שירותים ברחבי המערכת?
- כיצד מתבצע ניטור ואכיפת רמת השירותים?

- כיצד לוקחים אחריות על השותפים במערכת ועל התקשורת בתוך הארגון הווירטואלי?
- כיצד מאורגנים קבוצות והיררכיות השירותים במטרה לספק סמנטיקת שירותים מודולרית ואמינה?
- כיצד הקישוריות של משאבי המידע מתבצעת לכדי חישוב וכיצד מתבצע ניהול וניטור של קבוצות השירותים?

5.1.3. יכולות הליבה בארכיטקטורת ה-OGSA

איור 11 וטבלה 2 מספקות רמה ראשונית של פירוק סט יכולות הליבה הפונקציונליות המרכיבות את ארכיטקטורת OGSA. כמו כן, ניתן תיאור ראשוני של יכולות הליבה והקשרים הבסיסיים ביניהם (ברמת השימוש ההדדי Use Relations).



איור 11 - יכולות ארכיטקטורת ה-OGSA [15]

יכולות הליבה (OGSA Core Capabilities)	תיאור קצר
Infrastructure Services	מתרכזת באספקת יכולות תקשורת בין מגוון רחב של משאבים ב-Grid, דוגמת: מחשבים, משאבי אחסון ויישומים שונים.
Execution Management Services	ביצוע יעיל של תהליכים מרכזיים בסביבת ה-Grid ותפעולם, דוגמת: Provisioning ו-Life cycle management של המשאבים בסביבת ה-Grid.
Data Management Services	שינוע המידע והנתונים למקום שהם דרושים בו, תוך תמיכה בתהליכי ההעסקה, ביצוע השאילתות, עדכוני מידע של הטרנספורמציה והיתוך המידע.
Security Services	אכיפת מדיניות אבטחת המידע (Federated Policy) בסביבת ה-Grid.
Resource Management Services	ביצוע ניטור, הקצאה, פריסה וקונפיגורציה של משאבים ב-Grid בהתבסס על QoS מוגדר ומוסכם.
Self Management Services	תמיכה בהשגת רמת השירות המוצהרת על-ידי ה-Grid דרך יכולות ניהול אוטונומיות של התשתית.
Information Services	קבל מידע על ה-Grid והמשאבים בו, לדוגמה סטטוס של המשאב ורמת זמינותו.
Context Services	מגדירה את המשאבים ואת מדיניות השימוש הרלוונטית למשתמשים שונים ב-Grid.

טבלה 2 - יכולות הליבה של ארכיטקטורת ה-OGSA [15]

5.2. ייעוד הארכיטקטורה

על מנת ליצור ארכיטקטורה שתתאים ליצירת סטנדרט אחיד לפיתוח היישומים, יש צורך להגדיר את תמצית הדרישות (פונקציונליות ואי-פונקציונליות) שעמם היא נדרשת להתמודד. לצורך הגדרת הארכיטקטורה בוצע תהליך מסודר של איסוף הדרישות, המתבסס על תיאור הרשימה, שיש בה 15 תסריטי שימוש (Use Cases), שמהם נגזרות הדרישות לארכיטקטורת ה-OGSA. תסריטי השימוש המוצגים כאן משקפים אוסף של תסריטים גם מעולם היישומים וגם מעולם התשתיות, כל זאת ביחס לשני העולמות - המסחרי והמדעי. רצ"ב תיאור התסריטים לדוגמה בטבלה 3.

תקציר	תסריט שימוש
Data centers will have to manage several thousands of IT resources, including servers, storage, and networks, while reducing Management costs and increasing resource utilization.	מרכז מחשוב מסחרי Commercial Data Center (CDC)
Enable accurate prediction of the exact location of severe storms based on a combination of real-time wide area weather instrumentation and large-scale simulation coupled with data Modeling.	חיזוי סערות מזג האוויר Severe Storm Modeling
Delivering an entertainment experience, either for consumption or Interaction.	יישום מקוון של מדיה ובידור Online Media and Entertainment
Defines a virtual organization devoted to fusion research and addresses the needs of software developed and executed by this community based on the application service provider (ASP) model.	תסריט שיתוף והיתוך מידע National Fusion Collaboration (NFC)
A service-based distributed query processor supporting the evaluation of queries expressed in a declarative language over one or more existing services.	תסריט שאילתות מבוזרות Service-Based Distributed Query Processing
Workflow is a convenient way of constructing new services by composing existing services. A new service can be created and used by registering a workflow definition to a workflow engine.	תסריט Workflow בתצורת ה-Grid Grid Workflow
Inserting a supply chain between the Grid resource	תיווך משאבי ה-Grid

owners and end users will allow the resource owners to concentrate on their core competences, while end users can purchase resources bundled into attractive packages by the reseller.	Grid Resource Reseller
Extends the CDC use case by emphasizing the plethora of applications that are not Grid-enabled and are difficult to change: e.g. mixed Grid and non-Grid data centers, and Grid across multiple companies. Also brings into view generic concepts of utility computing.	Grid בין ארגוני Inter Grid
Compared to the online media use case, this use case emphasizes a high granularity of distributed execution.	Grid אינטרקטיבי Interactive Grids
Extends the use of Grids to small devices—PDAs, cell phones, firewalls, etc.—and identifies a set of essential services that enable the device to be part of a Grid environment.	Grid נייך Light Grid
A VO gives its members access to various computational, instrument-based data and other types of resources. A Grid portal provides an end-user view of the collected resources available to the members of the VO.	פורטל לארגון הווירטואלי - VO Virtual Organization (VO) Grid Portal
Preservation environments handle technology evolution by providing appropriate abstraction layers to manage mappings between old and new protocols, software and hardware systems, while maintaining authentic records.	ארכיב Persistent Archive
Refines the CDC and NFC use cases by introducing the scenario of the job submitter authorizing the resource on which the job will eventually execute.	הרשאה משותפת Mutual Authorization
Facilitates the mediation of resource usage metrics produced by applications, middleware, operating	שירות השימוש במשאבים Resource Usage Service

systems, and physical (compute and network) resources in a distributed, heterogeneous environment.	
Job execution, cycle sharing and provisioning scenarios.	ניהול תשתיות IT IT Infrastructure and Management*
Peer-to-Peer PC Grid computing, file sharing and content delivery scenarios.	תסריטי יישום אפליקטיביים Application Use Cases*
User-centered, contextualized and experiential-based approaches for ubiquitous learning in the framework of a Virtual Organization.	ה-Grid הלומד The Learning Grid
A distributed collaborative environment for developing and running simulations across administrative domains.	סימולציות HLA מבוזרות HLA-based Distributed Simulation*
An infrastructure for Application Service Provision (ASP) supporting different business models based on Grid technology.	תשתית ASP בתצורת Grid Grid based ASP for Business
Grid monitoring system scalable across wide-area networks and able to encompass a large number of dynamic and heterogeneous resources.	ארכיטקטורה לניטור Grid המחשוב Grid Monitoring Architecture

טבלה 3 - דוגמאות לתסריטי השימוש [15]

על מנת לראות איך ייעוד הארכיטקטורה והדרישות ממנה באות לידי ביטוי ומימוש ביכולותיה ראה [נספח ד' – יכולות ארכיטקטורת ה-OGSA](#).

5.3. קישוריות ותמיכה בסביבות הטרוגניות דינמיות

חלק מתסריטי השימוש שתוארו בהקדמה לסעיף זה מאופיינים בפרופיל של סביבות מחשוב הומוגניות, אשר רכיביהן מסוגלות לתקשר בצורה טובה. יחד עם זאת, ברור לחלוטין שברוב המקרים סביבת ה-Grid נוטה להיות הטרוגנית, מבוזרת, ובמקביל להקיף מגוון רחב של סביבות פיתוח והרצה ליישומים המבוזרים. לדוגמה (NET, JAVA), מערכות הפעלה, כגון: UNIX, Linux, Windows, ומערכות embedded, התקנים ותשתיות כגון: Computers,

Instruments, Sensors, Storage Systems, Databases Networks ומגוון רב של שירותים, אשר מסופקים על-ידי יצרנים שונים. כמו כן, סביבות מחשוב בתצורת ה-Grid נוטות להאריך ימים ולהיות דינמיות, ולכך עלולות להתפתח לא לפי התכנון ההתחלתי.

מכאן, למעשה, נובעת הדרישה מארכיטקטורת ה-OGSA לאפשר את הקישוריות בין כל המגוון הרב והמשתנה, ההטרוגני והמבוזר של המשאבים והשירותים ולהפחית את רמת הסיבוכיות בניהול סביבת המחשוב ההטרוגנית.

כיוון שנדרשות פונקציות רבות בסביבות המבוזרות, כגון אבטחת המידע או, לחילופין, ניהול המשאבים, הממומשות לרוב בצורה יציבה ואמינה במערכות ה-legacy הקיימות, וכיוון שאסור להחליפן או לשכתבן, הדרישה הבסיסית נובעת מכך שעל הארכיטקטורה לקשר בין משאבים ופונקציות אלה אל ה-Grid. המחשוב מתבצע במודל אחיד וסטנדרטי.

הצורך הבסיסי לתמוך במערכות הטרוגניות כחלק מ-Grid המחשוב מוביל אותנו לעמוד בדרישות הבאות כחלק מהארכיטקטורה:

• **וירטואליזציה של המשאבים (Resource Virtualization):** נדרשת הפחתת מורכבות

הניהול של מערכות מבוזרות לצורך ניהול מגוון ורחב של סוגי המשאבים (יישום, תשתיות ואחרים) בצורה ובסטנדרט אחידים.

• **יכולות ניהול בסיסיות (Common Management Capabilities):** הפשטת הניהול של

מערכות הטרוגניות דורשת מנגנונים עקביים שמאחדים את שיטת הניהול של המשאבים. נדרשות מינימום סט של יכולות ניהול שכיחות.

• **יכולת חיפוש וגילוי המשאבים ברשת (Resource discovery and query):** נדרשים

מנגנונים לאיתור משאבים בעלי תכונות שונות, על מנת לקבל את המאפיינים השונים שלהם, שעליהם לנהל סביבות הטרוגניות בצורה דינמית.

• **פרוטוקולים וסכמות סטנדרטים (Standard protocols and schemas):** בעלי

חשיבות גבוהה לקישוריות. השימוש בפרוטוקולים סטנדרטיים מקל ומאפשר את המעבר לשימוש ב-Grid.

5.4. שיתוף משאבים בין-ארגוניים

ה-Grid אינו מערכת, אלא מורכב בדרך כלל מאוסף משאבים הנמצאים בבעלות והנשלטים בידי ארגונים שונים. אחת היכולות המרכזיות של Grid המחשוב וארכיטקטורת ה-OGSA היא לתמוך בשיתוף השימוש במשאבים השונים הנמצאים בתוך הארגון ובין הארגונים השונים השותפים לו ובהגברתו.

ארכיטקטורת ה-OGSA נדרשת למענה במנגנונים הנדרשים, על מנת לספק קונטקסט ולקשרו למשתמשים, לבקשות, למשאבים, למדיניות ולהסכמים, בתוך הארגונים השונים המחוברים ל-Grid וביניהם.

שיתוף משאבים בין ארגוניים משפיע באופן ישיר על מגוון הדרישות בתחום אבטחת המידע, נושא אשר יורחב בהמשך.

דרישות לשיתוף המשאבים במסגרת ארכיטקטורת ה-OGSA כוללות:

- **תמיכה במרחב כתובות גלובלי (Global Name Space):** על ישויות המוגדרות ב-OGSA ליצור גישה לישויות האחרות בצורה שקופה תחת מגבלות אבטחת המידע, על מנת להקל על התהליך, כל זאת ללא קשר למיקום הפיזי.
- **תמיכה בשירותי מטה-מידע (Metadata Services):** מאפשר איתור, הפעלה ומעקב אחר ישויות. הארכיטקטורה אמורה לתמוך בגישה, בהפצה, בהעברה, באיחוד מטה-המידע של הישויות לרוחב מרחבי ניהול שונים בארגון ובין הארגונים השונים.
- **תמיכה באוטונומיה של אתרי המחשוב (Site Autonomy):** נדרשים מנגנונים שיאפשרו גישה למשאבים בין אתריים, וזאת תוך כדי כיבוד מדיניות אבטחת המידע שבהם.
- **תמיכה במידע לגבי השימוש במשאבים (Resource Usage Data):** נדרשים מנגנונים וסכמות סטנדרטיות לצרכי איסוף נתוני השימוש במשאבי ה-Grid בתוך הארגונים וביניהם והחלפתם, לצרכי ביצוע החיוב, הרישום של צריכת המשאבים על-ידי המשתמשים ב-Grid.

5.5. אופטימיזציה (Optimization)

הדרישה לאופטימיזציה בארכיטקטורה ה-OGSA מתייחסת לטכניקות, אשר משמשות לביצוע הקצאת השירותים והמשאבים, גם מבחינת הספק וגם מבחינת הלקוח. דוגמה נפוצה לאופטימיזציה של הספק קשורה לחוסר הקצאת משאבים במקרי קיצון (בעומסי עבודה גבוהים) מוביל לניצול מועט של המשאבים. ניצולם הרב יכול להשתפר על-ידי הקצאה גמישה המבוססת על המדיניות, כגון הזמנה מוקדמת של המשאבים למשך זמן מוגדר מראש. על אופטימיזציה על-פי דרישה לנהל מגוון רחב של עומסי עבודה, הכולל את הדרישה לפזרם, כי הם יכול להיות שיהיה קשה לחזות אותם. נוסף לכך, חשובה מאוד היכולת להתאימם באופן דינמי ולבצע תיעדוף, וזאת במטרה להשיג את יעדי השירותים. מנגנונים לביצוע מעקב אחרי ניצולת המשאבים, כולל מדידה וניטור אחרי הקצאתם ותחזית השימוש בהם על-פי דרישה, בדרך להשגת האופטימיזציה המבוססת עליה.

5.6. הבטחת רמת השירות (QoS)

על השירותים השונים, דוגמת ניהול והפעלת ג'ובים (Job Execution Services) ושירותי המידע (Data Services), לספק רמת שירות (QoS) נאותה, כפי שהוסכם עליה מראש. קריטריונים להבטחת רמת השירות אינם מוגבלים לזמינות, לאבטחת המידע ולביצועים בלבד. על הגדרת רמת השירות הנדרשת להתבצע באמצעות קריטריונים הניתנים לכימות ולמדידה.

הדרישות להבטחת רמת השירות כוללות:

- **חוזה רמת השירות (Service Level Agreement):** רמה זו מיוצגת על-ידי חוזים אשר מושגים בעזרת משא ומתן בין ספק השירות לצרכן השירות, וזאת עוד לפני שהשירות הופעל הלכה למעשה. המנגנונים נדרשים במטרה לייצר את החוזים האלה ולנהלם.
- **שמירה על רמת השירות (Service Level Attainment):** אם חוזה השירות מגדיר את הצורך לממש רמת שירות מובטחת, על המשאבים, אשר השירות משתמש בהם, להתאים את עצמם כך שרמתו תישמר. לשם כך, על הארכיטקטורה לספק מענה ומנגנונים לפעולות הניטור האלה: רמת השירות, הערכת ניצולת המשאבים ותכנון המשאבים הנדרשים וכוונותם.
- **הגירת השירותים (Migration of Executing Services):** דרישה זו אמורה לאפשר את ביצוע ההגירה של השירותים או האפליקציות, במטרה להתאים את עומסי העבודה לצרכי השגת יעדי הביצועים והזמינות הנדרשים.

5.7. הפעלת ג'ובים (Jobs Execution)

על ארכיטקטורת ה-OGSA לספק את מנגנוני הניהול לביצוע המשימות (Jobs), אשר מוגדרת לאורך מחזור חייהם בידי המשתמש. פונקציות, כגון תזמון, צפייה, שליטה וניהול שגיאות חייבות להיתמך על ידי הארכיטקטורה, אפילו כאשר משימה (Job) מבוצעת על גבי מספר משאבים הטרוגניים שונים ורבים.

דרישות הארכיטקטורה בנושא ביצוע המשימות (Jobs Execution) כוללות:

- **תמיכה בסוגים שונים של משימות (Support for various Job types):** תמיכה בהפעלת כמה סוגים של משימות, כולל משימות פשוטות ומשימות מורכבות, כגון תהליכים וזרימת העבודה או קומפוזיציות השירותים.
- **ניהול המשימה (Job Management):** זוהי דרישה חיונית. היכולת לנהל משימות לאורך משך חייהן היא קריטית. יחידת העבודה תספק ממשק ניהולי, שיאפשר את ניהול הקבוצות המורכבות של המשימות, דוגמת Job Arrays, Workflows וכיוצא בזה. מנגנונים, אשר יאפשרו שליטה בצעדים השונים המרכיבים את משימת העבודה, וכמו כן, ביצוע כוראוגרפיה ותזמון של השירותים הנדרשים, גם הם כחלק מארכיטקטורת ה-OGSA.
- **תזמון (Scheduling):** היכולת לתזמן ולבצע משימות על-פי עדיפות והקצאת המשאבים הנדרשת, לממש מנגנונים לתזמון מעבר לגבולות הארגונים ושימוש במגוון מנגנוני התזמון.
- **תחזית המשימות (Resource Provisioning):** לצורך ביצוע האוטומציה לתהליכים המורכבים של הקצאת המשאבים (Resource Allocation), התקנתם (Resource Deployment) וקונפיגורציה (Resource Configuration). יש צורך לאפשר את התקנת

האפליקציות והמידע הנדרש על גבי המשאבים ב-Grid ולקנפג (להפוך את התוכנה לתצורה הרצויה) אותם בצורה אוטומטית. אם נידרש להתקין ולקנפג סביבות אירוח, כגון מערכות הפעלה ותוכנה (Middleware), במטרה להכין את הסביבה הנדרשת לריצת המשימות - יש צורך לספק את מגוון סוגי המשאבים הנדרשים לביצוע המשימה.

5.8. שירותי הנתונים והמידע (Data Services)

הצורך בגישה יעילה לנתונים ושינוע כמויות נתונים גדולות הופכים לרווחיים יותר ויותר בתחומי המחקר המדעי והטכנולוגי. הצורך הגובר וההולך בשיתוף הנתונים, היכולת לאפשר גישה למידע שמור בבסיסי נתונים שונים, המנוהלים והמתוחזקים בצורות שונות ובנפרד זה מזה, הצורך בניהול יעיל של ארכיון המידע, מחייבים לתת מענה כחלק מארכיטקטורת ה-OGSA לנושא של שירותי הנתונים, אשר יתמקדו במענה לדרישות אלה:

- **גישה לנתונים ולמידע (Data Access):** גישה נוחה ויעילה לכמה סוגי מידע, כגון בסיסי נתונים בקבצים ובזרמי מידע (streams), באופן עצמאי וללא קשר למיקום הפיזי של הפלטפורמה, וזאת על-ידי הפשטה של התשתיות הנדרשות. נדרשים מנגנונים לשליטה וזכויות גישה למידע וברמות שונות של בידוד.
- **עקביות הנתונים והמידע (Data Consistency):** על הארכיטקטורה לספק מנגנון המבטיח שעקביות המידע תישמר כאשר מעדכנים או מעתיקים אותו.
- **התמדת הנתונים והמידע (Data Persistence):** על המידע ומידע-העל להיות תקפים לכל אורך חייהם. יש לאפשר את יישומם של כמה מודלים הנוגעים להתמדת זרימת המידע (persistence).
- **אינטגרציה של הנתונים והמידע (Data Integration):** הארכיטקטורה אמורה לספק מנגנונים לביצוע הקישוריות, החיפוש, על גבי מידע מבוזר והטרוגני. יש לאפשר את הדרישה לחפש מידע זמין בפורמטים שונים, אך בצורה אחידה.
- **ניהול מיקום הנתונים והמידע (Data Location Management):** על המידע להפוך לזמין במקום המיועד/ שזקוקים לו. הארכיטקטורה אמורה לאפשר בחירה בכמה דרכים, כגון העברה, העתקה ושמירה על-פי סוג המידע.

5.9. אבטחת המידע (Security)

ניהול בטוח של סביבת ה-Grid מצריך יכולת לשלוט ולנהל את הרשאות הגישה (Authorization) לשירותים ב-Grid, תוך כדי שימוש בפרוטוקולים אמינים של אבטחת המידע, המיושמים על-פי מדיניות אבטחת המידע של הארגון הרלוונטי. לדוגמה, הפצה והתקנה של אפליקציות תוכנה בתוך מערכת ה-Grid עלולה להצריך הזדהות והרשאות. מעבר לכך, ברור כי שיתוף המשאבים על-ידי משתמשים מצריך מנגנוני בידוד (Isolation). כמו כן, יש צורך במנגנונים אמינים וסטנדרטים, אשר ניתנים לפריסת המשאבים והיישומים ב-Grid

באופן מאובטח לרוחב של מרחבי ניהול (Administrative Domains) שונים, על מנת להבטיח את ניהולם באופן מבוטח.

דרישות אבטחת המידע בארכיטקטורת ה-OGSA כוללות:

- **הזדהות וניהול ההרשאות (Authentication & Authorization):** יש דרישה למנגנוני הזדהות (Authentication Mechanism) על מנת להגדיר ולזהות משתמשים ושירותים. על ספקי השירותים ב-Grid ליישם את מנגנוני ניהול ההרשאות (Authorization Mechanisms) לצורך הגדרת מדיניותם ואכיפתם. מדיניות זו תגדיר למי יש הרשאה לעשות שימוש באיזה משאב. על כלל המשתמשים ב-Grid המחשוב, להיות כפופים למדיניות אבטחת המידע שהוגדרה במרחבי ה-Grid, תוך מימוש יכולת ההזדהות, ניהול ההרשאות ושאר מרכיבי מדיניות אבטחת המידע שהוגדרה. על מנגנוני ניהול הרשאות הגישה ב-Grid המחשוב להיבנות על מנת להתאים את עצמם למגוון המודלים לניהול הרשאות הגישה למשאבים (Access Control Models) וליישומים שונים של מודלים אלה.
- **מגוון תשתיות אבטחת המידע (Multiple Security Infrastructures):** ארכיטקטורה בעלת אופי מבוזר, כמו OGSA, מעלה את הצורך לחבר ולקשר כמה תשתיות אבטחה. הדרישה המרכזית ממנה היא ניצול מקסימלי של תשתיות האבטחה הקיימות, תוך השתלבות בארכיטקטורות ובמודלים של אבטחת המידע הנמצאים בארגון.
- **פתרונות אבטחה היקפיים (Perimeter Security Solutions):** ב-Grid המחשוב שכיח מצב שבו נדרשת גישה למשאבים על-ידי גורמים מחוץ לגבולות הארגון. דרישה בסיסית בארכיטקטורה היא מנגנוני אבטחה וסטנדרט אבטחתי, אשר אפשר לפרוס אותם במטרה להגן על הארגון. וזה, במקביל לניהול אינטרקציות, גם בתוך גבולות הארגון וגם מחוצה לו. כל זאת מבלי להתפשר על מנגנוני האבטחה המקומית, כגון FIREWALL ומדיניות ה-INTRUSION DETECTION POLICIES.
- **בידוד (ISOLATION):** יש צורך במנגנוני בידוד, אשר יאפשרו את בידודם של המשתמשים, את הביצועים ואת התוכן, אשר מוצע בתוך Grid המחשוב.
- **האצלת הסמכויות (Delegation):** יש צורך במנגנונים לביצוע האצלת הסמכויות למתן הרשאות גישה מצרכני שירותים לספקיהם. על המנגנונים להיבנות לצורך מזעור הסיכונים הנובעים מן היכולת לבזר את הרשאות הגישה למשאבים, על-ידי בנייה ומימוש היכולת להעניק אותן להם בהקשר של משימה או פרק זמן מוגדר.
- **חלופת מדיניות אבטחת המידע בין גורמים שונים (Security Policy Exchange):** נדרשים מנגנונים שיאפשרו למבקשי השירותים ולספקי החליף את מדיניות האבטחה בצורה דינמית, וזאת בכדי להיות מסוגלים לבצע משא ומתן בנושא האבטחה ביניהם.

- **זיהוי חדירה, הגנה והזדהות מאובטחת (Intrusion Detection, Protection & Secure Login):** יש דרישה לניטור יעיל כדי לזהות חדירות, זיהוי שימוש בלתי הולם במשאבי ה-Grid, פעולות זדוניות, וירוסים או התקפות. כמו כן, יש להגן על אזורים חשובים או פונקציות על-ידי הרחקת ההתקפה מאזורים אלה.

5.10. הפחתת עלויות הניהול (Administrative Cost Reduction)

מורכבות הניהול של מערכות מבוזרות ורחבות היקף בסביבות הטרוגניות מגביר את עלויות התפעול ואת הסיכון בטעויות אנוש. תמיכה במשימות הניהול של מערכות אלו על-ידי תמיכה קונסיסטנטית באוטומציה של תהליכי הניהול וביצוע ניהול עקבי של אוסף המשאבים דרך וירטואליזציה של משאבי ה-Grid. כל אלה מהווים דרישה בסיסית מן הארכיטקטורה.

דרישות ניהול והקטנת עלויותיו כוללות:

- **ניהול מבוסס מדיניות (Policy-based management):** לצורך ביצוע האוטומטיזציה של

משימות הניהול במערכות ובסביבות המחשוב של ה-Grid, זאת על מנת שפעילות ה-Grid ומערכותיו תתאים למטרות הארגון. כל זאת מתנהל גם ברמה הנמוכה של המדיניות, אשר שולטת באספקטים, דוגמת הצורה שבה המשאבים מנוטרים ומנוהלים וגם בהיבטי הניהול ברמה הגבוהה יותר, דוגמת בקביעה ובניהול מדיניות בנושאים, כמו: איך וכיצד תהליכים עסקיים מנוהלים וכיצד מחייבים בגינם? ניהול מבוסס מדיניות הוא דרישה בסיסית, אשר יש ליישמה בכל רמת מערכת. אספקטים של המדיניות כוללים נושאים, דוגמת זמינות, ביצועים, אבטחה, תזמון ותיווך.

- **ניהול תכנים אפליקטיביים (Application contents management):** דרישה למנגנוני

ניהול תוכן אפליקטיבי, שיכולים לסייע בפריסה, בקונפיגורציה ובתחזוקת מערכות מורכבות, וזאת על-ידי מתן האפשרות להגדיר את המידע הרלוונטי באפליקציות כיחידה לוגית אחת ולנהלם. גישה זו מאפשרת למנהלי המערכת לתחזק את רכיבי האפליקציות בצורה תכליתית ואמינה, וזאת גם ללא ידע נרחב עליהם.

- **מנגנוני איתור הבעיות (Problem Determination):** דרישה למנגנוני איתור התקלות,

וזאת כדי שמנהלי המערכת יוכלו לזהות ולהתמודד בצורה מהירה עם בעיות דחופות.

5.11. יכולת גידול (Scalability)

מערכות ה-Grid בעלות הממדים הגדולים מסוגלות ליצור ערך מוסף. נושאים, כגון הפחתת משמעותית בזמן הביצוע של משימות רלוונטיות, תוך יכולת לעשות שימוש מנוהל במספר רב של משאבים, ובכך לאפשר שירותים חדשים, שלא היו אפשריים עד כה. מערכות אלו, למרות זאת, עלולות ליצור/ לגרום לבעיות, וזאת משום שהן מציבות דרישות רציניות, בעיקר בנושא תשתיות הניהול.

- **ארכיטקטורת הניהול (The management architecture):** על ארכיטקטורת הניהול הנמצאת באפשרות לגדול לאלפי משאבים במגוון וברוחב. הניהול אמור להתבצע בצורה היררכית או בין נקודה לנקודה בצורה מבוזרת.
- **מחשוב בהספק גבוה (High-throughput computing):** עבור התאמה ואופטימיזציה של משימות (Jobs), המבוצעות בצורה מקבילית, יש דרישה למנגנונים ולתשתיות מחשוב בעלי הספקי מחשוב גבוהים. זאת במטרה לשפר את ההספק הכולל של כל התהליך החישובי ובפרט את האופטימיזציה של יחידת מחשוב בודדת.

5.12. זמינות (Availability)

זמינות גבוהה של מערכת המחשוב מושגת בדרך כלל על-ידי חומרה יקרה בעלת סבלנות גבוהה לתקלות או על-ידי מימוש מערכת אשכול (cluster) מורכבת. משום שמערכות IT מסוגים שונים ורבים מספקות תשתית ציבורית חיונית, מספרם של המערכות הנדרשות לתפקד בזמינות גבוהה הולך וגדל.

תודות לכך שטכנולוגיות ה-Grid מאפשרות גישה נוחה למגוון רחב יותר של מאגרי משאבים, בין ארגונים ובתוך הארגונים, אפשר להשתמש בה כאבן בניין למימוש סביבת ריצה אמינה ובעלת זמינות גבוהה מאוד.

יחד עם זאת, חשוב להבין כי בגלל ההטרוגניות של ה-Grid והשימוש ברכיבים יום-יומיים, אשר בדרך כלל אינם קיימים במערכות בעלות זמינות גבוהה, כל אלה גורמים לבעיות קשות בהשגת רמות הזמינות הדרושות.

בסביבה מסובכת כל-כך, מימוש העקרונות, דוגמת ניהול מבוסס מדיניות ושליטה אוטונומית, מהווים את המפתח למימוש מערכות בעלות רמת גמישות ויכולת התאוששות גבוהה.

• **התאוששות מאסון (Disaster recovery):** דרישה למנגנונים יעילים להתאוששות

מתקלות, שבאה להבטיח כי התפקוד של מערכות ה-Grid יוכל להציע התאוששות מהירה ויעילה במקרה של אסון שנגרם באופן טבעי או דרך טעות אנוש, והימנעות ממצב של אי-זמינות השירותים לטווח הארוך. כמו כן, הארכיטקטורה מגדירה דרישה ליכולת גיבוי מרוחק והפשטה או אוטומציה של תהליכי ההתאוששות.

• **ניהול תקלות הכשלים (Fault management):** דרישה למנגנוני ניהול התקלות.

מנגנונים אלה אמורים לעזור בעת ריצת המשימות ב-Grid ולהבטיח את המשך פעולתם, למרות התקלות שעלולות להתרחש במשאבים. כמו כן, מוגדרת דרישה למנגנוני הניטור, איתור התקלות ואבחון שורש הבעיה או ההשפעה על המשימות הרצות, בנוסף לאוטומציה של טיפול בתקלות ושימוש בטכניקות, כגון Checkpoint Recovery.

5.13. נוחות השימוש והתרחבות (Ease of Use and Extensibility)

מבחינת המשתמש, הארכיטקטורה אמורה לאפשר למסך את מורכבות סביבת המחשב. היא דורשת כלים וסביבות זמן ריצה, המאפשרים לנהלם בעבורו ולספק הפשטה שימושית ברמה הנדרשת.

יחד עם הצורך למיסוד והפשטה, עלינו לזכור כי יש "משתמשי-על" [Power Users] בעלי דרישות אפליקטיביות רחבות יותר, אשר צריכים ודורשים את היכולת לקבל החלטות ברמה הנמוכה ולתקשר עם המערכות גם כן ברמה זו. לכן חובה להשאיר את אפשרות הבחירה

למשתמש הקצה שיבחר את רמת ההתקשרות הנדרשת עם סביבת ה-Grid.

כיוון שאין אפשרות לחזות את כל סוגי הצרכים שיש למשתמשים, על מנגנוני קביעת המדיניות להיות ממומשים דרך המנגנונים, אשר ניתנים להרחבה או להחלפה. וזה, תוך כדי מתן האפשרות לארכיטקטורה להתפתח עם הזמן ולאפשר למשתמשים לייצר מנגנונים בעלי מדיניות המתאימה לצרכים המדויקים שלהם. נוסף לכך, על הרכיבים המרכזיים עצמם לאפשר הרחבה והחלפה. יכולת התרחבות כזו תאפשר לצד שלישי או לפיתוח מקומי לממש ולייצר שירותים בעלי ערך מוסף.

6. מיפוי יכולות ה-ESB ליכולות ה-OGSA

בטבלה 4 מוצגים השירותים הבסיסיים שהגדרנו עבור ה-ESB הסטנדרטיים, ש-OGSA מגדירה והסטנדרטים ממשפחת ה-WS*, שמהם היא מקבלת את תמיכתה.

משפחת שירותים ב-ESB	סטנדרטים רלוונטיים ל-OGSA	סטנדרטים רלוונטיים ל-WS*
שירותי ביטחון מערכות המידע	Trusted Computing, Authorization, P2P and Firewall Issues נתינת מענה ל-Security עבור P2P ו-Firewalls	תוך התבססות על תקני WS*: WS-Security, WS-Trust, WS-Federation, SAML, WS-SecureConversation
שירותי שינוע המסרים		WS-Addressing, WS-MessageDelivery, WSRM (Reliable Messaging), (MOTM) Efficient Messaging, WS-Notification, WS-Eventing
שירותי אירוח שירותים	חשיפת ממשקי תשתית ה-Grid, Grid-RPC, תמיכה בטרנזקציות, שירותי Workflow	
שירותי ממשק המשתמש		WSRP (Remote Portals)
שירותי אחסון המידע	OGSA-DAI (Data Access & Integration), Grid-FTP, Data Replication, High-level Publish/Subscribe, ניהול האחסון, הגדרת מידע בינרי, ניהול הטרנזקציות	
שירותי שליטה ובקרה (ש"ב)	קונפיגורציית משאב/ שירות, ניהול פריסת השירות, ניהול מחזור חיי השירות, Grid Economy Model	תוך התבססות על תקני WS*: WSDM, WS-Management, WS-Transfer
שירותי גילוי		UDDI, WS-Discovery, WS-

Resource Framework (WSRF)		
	Virtual Organizations	שירותי שיתוף
תוך התבססות על תקני WS*: WS-Policy, WS-Choreography, WS-Coordination	תמיכה במערכות Legacy, טרנספורמציות המידע	שירותי תיווך

טבלה 4 - כיסוי משפחות השירותים על-ידי OGSA

כפי שאפשר לראות בטבלה 4 לעיל, יש כיסוי מלא של שירותי הבסיס על-ידי WS* ו-OGSA. מעבר לכך יש תחומים נוספים המכוסים על-ידי OGSA, כגון: Job, System Metadata, Scheduling and Execution Management. כמו כן, יש תמיכה מובנית ב-WS-Policy, WS-Agreement על מנת לתמוך ב-SLA בין השירותים והתמיכה בקבלת מידע על העומסים ברשת.

כל השירותים המוגדרים ב-OGSA הם Grid Service - שירות המבוסס על Web Service, אך תחת מוסכמות המגדירות כיצד צרכן השירות ניגש ומשתמש בשירות. ארכיטקטורת ה-OGSA מגדירה את המכניזם שבעזרתו אפשר ליצור ולנהל את Grid Services וגם את המכניזם שדרכו עובר המידע ביניהם.

פרויקט ה-GT4 Globus Toolkit (GT4), המוביל כיום במימוש ארכיטקטורת ה-OGSA, מכיל כבר רכיבים רבים המהווים את מרבית היכולות הנדרשות ממשפחות השירותים (אפשר למצוא הרחבה על שירותי ה-GT4 בנספח).

בטבלה 5 מצוינים הרכיבים, אשר נמצאים בפרויקט ה-GT4 ומממשים חלק מן הסטנדרטים של OGSA (כפי שהם מופיעים בטבלה 4). גם כאן יש מיפוי למשפחות השירותים המרכזיים ב-ESB, אשר אפשר לבססם על גבי רכיבים אלה.

משפחת השירותים	
מיפוי לרכיבים ב-GT4	ב-ESB
WS-Security, WS Authentication, WS Authorization, CAS, Delegation Services, Credential Management	שירותי ביטחון מערכות המידע
GT4 Core Services (WS-Addressing, WS-Notification)	שירותי שינוע המסרים
GT4 Core Services, GT4IDE, GPT (Globus Packaging Toolkit)	שירותי אירוח שירותים
OGCE (Open Grid Computing Environment)	שירותי ממשק המשתמש
OGSA-DAI (מימוש)	שירותי אחסון

Reliable File Transfer, Grid-FTP, Replica Location Service	המידע
ניטור : Monitoring & Discovery System (MDS4) ניהול : WSDM (בגרסאות הבאות)	שירותי שליטה (ובקרה) (שוי"ב)
Monitoring & Discovery System (MDS4)	שירותי גילוי
בסיס : Monitoring & Discovery System (MDS4)	שירותי שיתוף
OGSA-DAI, Grid Resource Allocation & Management חסר תמיכה ב-BPEL	שירותי תיווך

טבלה 5 - כיסוי משפחות השירותים על-ידי רכיבי ה-GT4

אפשר לראות כי OGSA מביאה עמה פתרונות רבים מעולם ה-Grid אל עולם השירותים, כאשר היא מתבססת על סטנדרטים קיימים (WS*), על מנת ליצור תשתית אפליקטיבית המאפשרת לחשוף את השירותים על גבי ה-Grid. בעזרת השימוש ב-OGSA אפשר לקבל את העמידות, את הסקלביליות, את הזמינות הגבוהה ואת השילוב של ביטחון מערכות המידע מבחינת כל הבחינות.

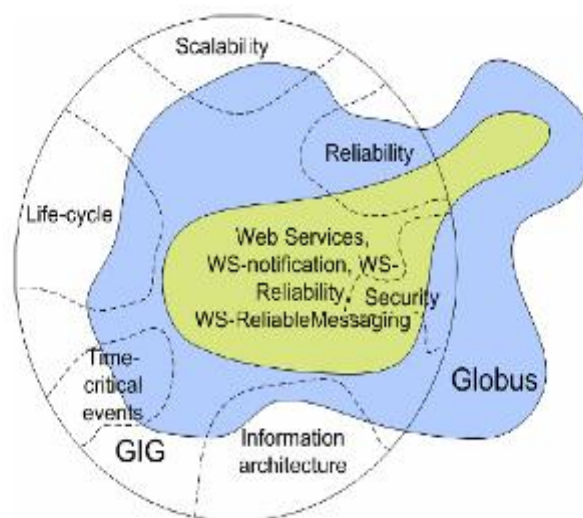


Figure 3 Overlap between the GIG/JBI, Globus, and Web Services standards.

איור 12 - החפיפה בין ה-Globus, GIG והסטנדרטים של Web Services [32]

איור 12 מציג באופן גרפי את המיפוי של ארכיטקטורת ה-GIG האמריקאי על-ידי הסטנדרטים של ה-WS* ופרויקט ה-Globus Toolkit. כפי שאפשר לראות יש כיסוי רב של דרישות ה-GIG ונגיעה בכל התחומים הנוגעים אליו של הרכיבים הקיימים ב-Globus Toolkit.

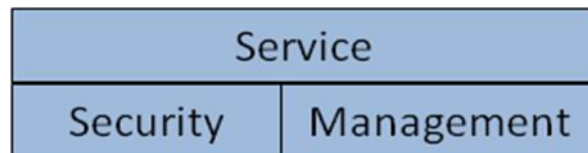
7. ארכיטקטורת ESB מבוססת על בסיס ה-OGSA

כפי שראינו בפרק הקודם יש רכיבי תוכנה רבים המסופקים באמצעות מימוש ה-GT4. בחלק זה ננסה להבין כיצד מממשים את ארכיטקטורת ה-ESB המבוססת, שתוארה על פני רשת המחשבים בעזרת GT4.

על מנת להגדיר את הארכיטקטורה המבוססת יש לציין אילו רכיבי תוכנה נמצאים על כל אחד מן המחשבים המשתתפים ברשת.

כפי שצוין לפני כן, OGSA מגדירה את השירותים כ-Grid Services. שירות ה-Grid Service עוטף את ה-Web Service, מבוסס עליו ופועל תחת מוסכמות נוספות המגדירות כיצד צורך השירות משתמש בו? המוסכמות נוגעות לנושאים אשר צוינו לפני כן ואין לגביהם מענה מספק ב-Web Services. דוגמה בולטת היא הצורך ביכולת לתפעל משאב מסוים (יכול להיות חומרה, כגון מחשב או תוכנה, כגון שירות). כאשר משאב מיוצג בעזרת Grid Service נוספות לו יכולות המאפשרות לנו להתחיל/ להפסיק את פעילותו ולאתחלו.

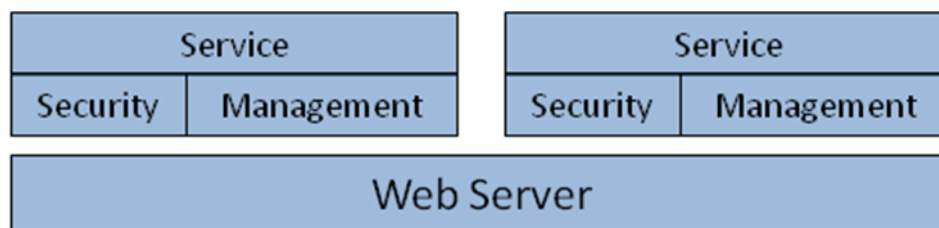
כל Grid Service, המיוצר בעזרת GT4, מממש גם את הממשק העסקי של Web Service (היכולות האופרטיביות) וגם את הממשק הטכני (כמו יכולות ביטחון מערכות המידע או ניטור), כאשר המימוש הטכני מבטיח לנו עמידה בסטנדרטים המצוינים לעיל. יתרון משמעותי, הנובע מאופן המימוש, הוא הביזור המוחלט של כל השירותים. אפשר להפעיל על המחשב (מרוחק ככל שיהיה) כל שירות ועדיין יהיה אפשר לשלוט על התנהגותו.



איור 13 - Grid Service (העוטף את ה-Service)

באיור 13 אפשר לראות כי Grid Service עוטף Service כלשהו (המתואר ב-Web Service) בעזרת Security (סטנדרטים ממשפחת WS-Security המוזכרים לעיל) ובעזרת Management (סטנדרטים ממשפחת WSDM).

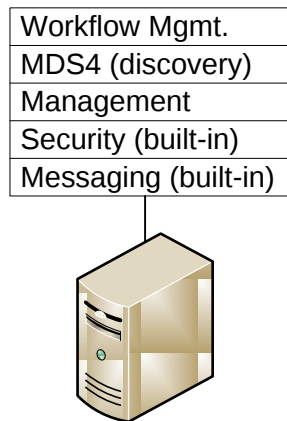
כעת, אפשר לקחת את ה-Grid Service ולהתקין אותו על מכונה כלשהי. על המכונה אמור לרוץ HTTP engine/ Web server (כגון Apache) המסוגל להציג Web Service (רגיל) למשתמש חיצוני. על המכונה (וה-Web server) אפשר להריץ כמה שירותים במקביל (כפי שמוצג באיור 14).



איור 14 - שני Grid Services הרצים על Web Server אחד

כלי GT4 מספק לנו כמה שירותים (Grid Services) מוכנים. שירותים אלה מממשים חלק ממשפחות ה-ESB הנ"ל. שירות חשוב המסופק לנו על-ידי GT4 הוא MDS4 (Monitoring & Discovering System), המאפשר לרשום למאגר המידע שירותים אחרים. היכולת הזו מאפשרת לשירות להירשם במאגר מידע מבוזר ולאחר מכן יהיה אפשר לחפש אותו ולהשתמש בו. מאגר המידע מאפשר לנו לקבל מידע לגבי השירות הרשום אצלו ובעיקר את מיקומו. במאגר המידע אפשר להשתמש גם במהלך הפיתוח וגם במהלך הריצה. כאשר מפתחים שירות חדש ורוצים שהוא ימופה לשירות קיים במיקום מוגדר וקבוע, אפשר למצוא אותו במהלך הפיתוח ולהגדירו (בצורה סטטית) בשירות החדש. כל פעם שנפעיל את השירות החדש הוא יפנה ישירות למיקום הקבוע של השירות הקיים וישתמש בו. אם נעביר את השירות הקיים למיקום אחר, נצטרך לעדכן את השירות החדש. דרך נוספת היא לקבוע כי השירות החדש ימופה במהלך ריצה (בצורה דינמית) לשירות הקיים, כלומר כל פעם שנפעיל את השירות החדש הוא ישתמש במאגר המידע על מנת לזהות את מיקום השירות הקיים ולהשתמש בו. בניית השירות בעזרת GT4 מאפשרת לנו את שתי הדרכים הנ"ל.

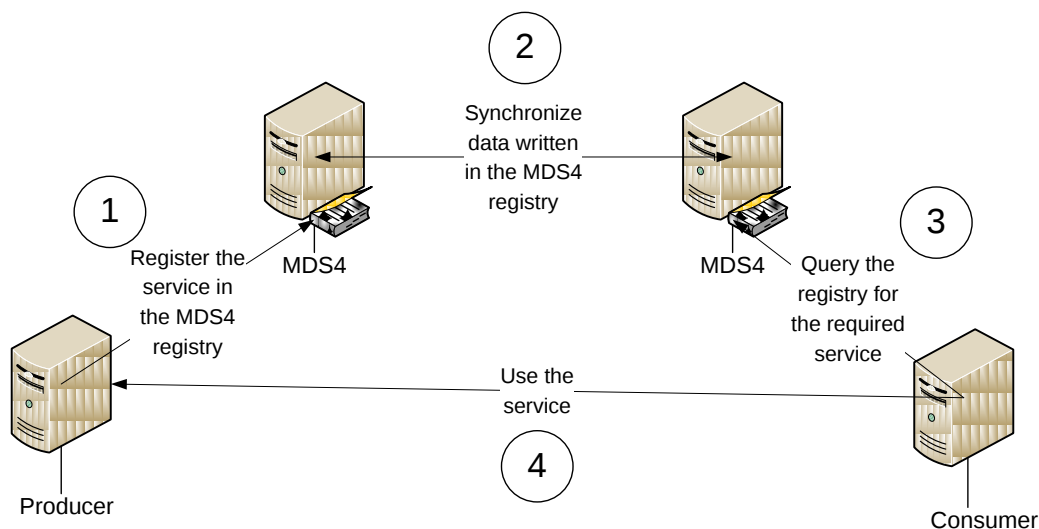
כפי שהוזכר, מאגר המידע הוא גם Grid Service, שבאפשרותו להימצא על כל מכונה. אם מחליטים לא להריץ את שירות מאגר המידע על מכונה מסוימת, אפשר להפסיק את פעילותו או אפילו לא להתקין אותו כלל. אם השירותים, הנמצאים על המכונה ירצו להשתמש בשירות זה, עליהם לדעת (באופן סטטי) את מיקום שירות מאגר המידע על מכונה אחרת, ואז יעשו בו שימוש. אפשר להגדיר למאגר המידע כתובת של מאגר מידע אחר, שבעזרתה הוא אמור לסנכרן מידע. כך נבנה לנו מאגר מידע מבוזר, המכיל מידע לגבי כל השירותים שנרשמו באחד מן המאגרים.



איור 15 - דוגמה לשרת המכיל כמה שירותים הניתנים על-ידי OGSA

דוגמה למימוש מענה תקשורת באמצעות OGSA אפשר לראות באיורים 15 ו-16. באיור 15 אפשר לראות את השירותים הניתנים על-ידי OGSA עבור כל אחד מלקוחות המערכת. בין שירותים אלה אפשר למנות את שירותי שינוע המסרים ואבטחת מידע (הנמצאים בכל Grid-service) ואת שלושת השירותים הנוספים: Workflow Mgmt – לניהול Workflow (Orchestration), MDS4 – לגילוי שירותים ומשאבים ו- Management – לניהול השירותים הנמצאים על השרת (עצירת/ אתחול השירות). באיור 16 ניתן האופן בו שני צרכנים מבזרים מתמפים זה מול זה על-ידי שימוש בספרייה הארגונית MDS4:

- (1) יצרן השירות נרשם באחד משירותי ה-MDS4 הידועים לו.
- (2) שירות ה-MDS4 מסונכרן עם שאר שירותי ה-MDS4 המחוברים אליו.
- (3) הצרכן פונה לשירות ה-MDS4 ומקבל ממנו מידע לגבי מיקום השירות שהוא מחפש.
- (4) הצרכן מפעיל את השירות אצל יצרן השירות.



איור 16 - תהליך מציאת השירות ב-ESB על בסיס תשתיות ה-Grid

8. מתודולוגיה, כלי הניתוח וחומרים

בפרק זה נסקור את המתודולוגיה שבאמצעותה נשווה בין החלופות השונות: חלופת ה- Web Services וחלופת ה- Grid Services. נגדיר את המתודולוגיה להשוואה, את המאפיינים האיכותיים לה בין החלופות השונות על בסיס תשע המשפחות של צבא ארה"ב ואת התרחישים על-פיהם נבצע אותה. תוצאות ההשוואה עצמה יוצגו בפרק הבא.

8.1. חלופות

במסגרת הניתוח של אפשרויות מענה ה-ESB, נבחן את שתי החלופות התפיסתיות המרכזיות הנמצאות בשוק:

1. שימוש ב- Web Services - סדרת פרוטוקולים סטנדרטיים המאפשרים תקשורת בין אפליקציות (API) על בסיס פרוטוקול HTTP (על-פי רוב), המשמש לתקשורת סטנדרטית באינטרנט.

2. שימוש ב- Grid Services – סדרת פרוטוקולים המבוססים ברובם על Web Services והמאפשרים להתמודד עם נושאי איכות, שירות וביזור.

8.2. מאפיינים איכותיים להשוואה בין החלופות

על מנת להשוות בין שתי החלופות יש להגדיר סדרה של מדדי השוואה בין החלופות. מדדים אלה יתבססו על השוואת מימוש של תשע משפחות השירותים ב-ESB על בסיס כמה תרחישים:

1. שירותי אבטחת מידע.
2. שירותי שינוע המסרים.
3. שירותי אירוח השירותים.
4. שירותי ממשק המשתמש.
5. שירותי אחסון המידע.
6. שירותי שליטה ובקרה.
7. שירותי גילוי.
8. שירותי שיתוף.
9. שירותי תיווך.

8.3. כלי לניתוח והשוואה

ניתוח הממצאים נעשה על-ידי ביצוע בחינת הארכיטקטורה של שתי החלופות : חלופה המבוססת על Grid Services ואחרת המבוססת על Web Services, תוך התבססות על תרחישי השימוש. הבחינה תכלול את מענה הארכיטקטורה לדרישות, והפערים הנדרשים למימוש. על מנת לבחון בצורה נאותה את התאמת הארכיטקטורות למציאות, נדרש לבחור תרחישי שימוש שייצגו מודל עקרוני, בהן יידרש לעמוד הגוף העסקי.

8.4. מודל הבחינה וההשוואה

Use Case Maps (UCM) היא שיטת תיאור יכולות והתנהגות המערכת, אשר בעזרתה אפשר לקחת את ה-Use Cases (UCs) וליצור את הקשר הסיבתי (causality) בין ה-UCs. Use Cases - מסמכים, אשר כל אחד מהם מתאר פונקציונליות ספציפית של המערכת ואת האינטרקציה שלה עם הסביבה [33]. הרעיון הוא לזהות, מתוך מסמכי הדרישות, את כל השחקנים (Actors) שמשתמשים במערכת ואת התרחישים שהם מפעילים עליה, וגם לפרט את האינטרקציה של השחקן מול המערכת. ה-UCs מתארים את האינטרקציה בשפתו של הלקוח ואין התייחסות למונחים הטכניים (ה-UCM מתרכז בהעברת ה"מה" על המערכת לעשות? ולא ב"איך" עליה לעשות?).

ה-UCM לוקח, למעשה, את רכיבי הבניין שלו (UCs), מחבר ביניהם על-ידי קו המתאר את התהליך ואת קשר הסיבה והתוצאה בין ה-UCs. נוסף לכך, הוא מאפשר לאגד תהליך כזה וליצור רמת שימוש נוספת העוטפת את ה-UCs הפנימיים.

בעזרת UCM אפשר בעצם לראות את התמונה השלמה, אשר מחברת בין שלב דרישות המערכת לבין שלב האפיון. הרעיון הוא שאין צורך לחכות לשלב האפיון על מנת ליצור את ה-Sequence Diagrams (הבנויים על מחלקות מה-Class Diagrams), אלא כבר בשלב ה-High Level Design לחלק את המערכת לרכיבים ולהראות את הקשרים ביניהם. נוסף לכך, אפשר להראות את התהליכים הדינמיים המתבצעים במערכת, ובעיקר קל לראות את התהליכים המתבצעים במקביל. לכל תהליך יוגדרו תנאי הכניסה (pre conditions) ותנאי היציאה ממנו (post conditions), וגם אותם אפשר לראות ויזואלית ולעקוב אחר תקינות פעולת המערכת (מנקודת מבטו של הלקוח). במסגרת המחקר ביצענו שימוש ב-UCM navigator - כלי Open Source לבנייה ולתיאור של ה-UC's [33].

8.5. מודל ה-UCM

מתודת UCM מאפשרת ללקוח להבין כיצד המערכת תהיה בנויה (ארכיטקטורת-על של המערכת), וכיצד התהליכים יתבצעו? על מנת לכתוב UCM, עלינו להבין את דרישות המערכת ואת ה-UCM אפשר לבנות במקביל לבניית ה-UCs.

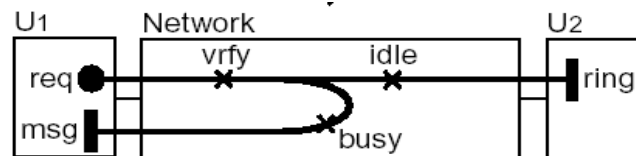
שימוש בשיטה זו יכול לגרום להבנה טובה יותר של המערכת ובעזרתה אפשר לבצע מידול תהליכים (גם תהליכים דינמיים, שהרכיבים בהם נקבעים במהלך ריצה) בהצלחה רבה. נוסף לכך, בעזרת

UCM אפשר לקחת את התנהגות המערכת המתקבלת ולהצביע על נקודות איכותיות שעליהן שמים דגש, ובכך לזהות את הנקודות העיקריות שלה.

הדוגמה הבאה מתארת מערכת טלפוניה, אשר בעזרתה לקוח כלשהו (U1) רוצה להתקשר ללקוח אחר (U2). לקוח U1 יוצר את הבקשה לשיחה (req עם עיגול המסמן התחלה – req מציין triggering event או pre-condition), מעביר את הבקשה לרכיב הרשת, שבו נמצא גורם האחראי (responsibility מסומן כ-X על הקו) על בדיקת הסטטוס הנוכחי של U2 (vrfy).

משם יש התפצלות, אם vrfy הצליח או נכשל. סיום התרחיש מסומן בקו (מציין resulting event או post-condition שאומר מהו סטטוס סיום הפעולה).

יש שלושה רכיבים המשתתפים בתהליך: רכיבי לקוח U1, U2 ורכיב רשת Network. יש קשר בין הרכיבים U1 ו-Network ובין Network ו-U2 (מציין על-ידי קו הנמצא בין הרכיבים). יש כמובן גם דרך לציין דברים נוספים בתהליך (כמו שגיאות).



בדוגמה מתואר תהליך יצירת הקשר ומצוינות שתי אלטרנטיבות: הצלחה ביצירת הקשר (ring אצל U2) או כישלון (msg אצל U1 – צליל תפוס).

8.6. מודל ההשוואה – ATAM

שיטת ה-ATAM (Architecture Tradeoff Analysis Model) [36] משמשת לצורך השוואה בין ארכיטקטורות, אשר מטרתה להוכיח את יעילותה של הארכיטקטורה הנבחרת לפתרון דרישות המערכת המסוימת. נוסף לכך, היא מציינת את הלבטים השונים בה ואת הנקודות, שבהן נעשה חלופות (tradeoff) בין האפשרויות השונות. רצוי להפעיל את השיטה כבר בשלבים הראשונים, על מנת לכסות את כל ההחלטות שנעשו בדרך ושכל בעלי העניין במערכת יסכימו כי זוהי הארכיטקטורה הרצויה.

הרעיון העיקרי מתבסס על זיהוי נקודות רגישות בפונקציונליות של המערכת והשוואה בין ארכיטקטורות שונות, על מנת למקסם את הרווח בפתרון הנבחר, ללא צורך במימוש וביישום מלא שלהן. השיטה אינה משווה רק את הדרישות הפונקציונליות של המערכת, אלא גם את הדרישות האיכותיות. בעזרת שיטה זו אפשר להשוות, לדוגמה את התגובה לשינויים הנובעים מן המערכת, את העמידה בזמנים שלה וכדומה.

השיטה מציינת את השלבים ומספקת דרך פעולה למציאת הנקודות הרגישות (sensitivity points) – תכונות של רכיב יחיד או הקשר בין הרכיבים, הקריטית להשגת מענה לדרישה כלשהי, עבורן יושקעו מחשבה וזמן, ואת נקודות האיזון במערכת (tradeoff points) – תכונה המשפיעה על יותר ממאפיין אחד, כלומר נקודות רגישות של יותר מדרישה אחת).

התוצאה הרצויה של השיטה היא דירוג התרחישים לפי עדיפות, הבנה ותייעוד של הארכיטקטורה המוצעת לפתרון (גם על-ידי הארכיטקטים וגם על-ידי בעלי העניין האחרים). בהקשר לשאלה

"למה?" נבחרה דווקא הארכיטקטורה המוצעת, כל נקודות הרגישות ונקודות "צוואר הבקבוק" שהתגלו ורשימת הסיכונים (החלטות ארכיטקטוניות בעיתיות) וחוסר-הסיכונים (החלטות ארכיטקטוניות הנובעות מהנחותיהן).

השיטה מגדירה את השלבים, שבהם כדאי לבחור על מנת לדון על הארכיטקטורה המוצעת ובכזו אשר כל הצדדים יסכימו עליה (הפותרת את הבעיות העיקריות שהמערכת אמורה לפתור).

הקלט ל-ATAM הוא מסמך הדרישות ותרחישי השימוש (Use Cases), שנובעים מן הדרישות (התרחישים הפונקציונליים, תרחישי הגדילה ותרחישי הביצועים), והפלט הוא תיאור הארכיטקטורה שנבחרה תוך כדי התבססות על מאפייני האיכות, שעליהם הושם דגש. שימוש ב-Use Cases מבהיר את התרחישים הפונקציונליים ומאפשר לעוסק ב-ATAM להבין טוב יותר את מטרת המערכת ואת הקשר בין הסוגים השונים שלהם.

השימוש במתודולוגיית ATAM נעשה דרך השלבים הבאים:

- הגדרת השחקנים ו/ או התהליכים המדויקים במערכת – הגדרת תרחישי שימוש פרטניים, כאשר כל אחד מהם מייצג התנהגות/ דרישה (אחת או יותר) של המערכת, שעמה הארכיטקטורה אמורה להתמודד ולה היא אמורה לתת מענה.
 - הגדרת היכולות הלא-פונקציונליות במערכת ומאפייני שימוש – הגדרת התרחישים ומדדי האיכות, שהארכיטקטורה נדרשת לתמוך בהם. תרחישים אלה מייצגים מאפייני עומס, גידול, זמינות ושרידות, שבהם המערכת נדרשת לעמוד.
 - הגדרת מאפיינים איכותיים להשוואה בין הארכיטקטורות ומציאתם – הגדרת סט קריטריונים איכותיים להשוואה בין הארכיטקטורות, תוך מיקוד בנקודות האיזון ובנקודות הרגישות שבהן עמידות הארכיטקטורה נבחנת.
 - עבודה המתבצעת בשלבים עם כל בעלי העניין במערכת, על מנת לקבל ארכיטקטורה, אשר ממקסמת את הרצונות מהמערכת, ולוודא שכולם מבינים אותה.
 - סיכום השוואתי בין הארכיטקטורות המוצעות, תוך דירוג איכות המענה לכל תרחיש וקריטריון שהוגדר. שלב זה מחייב יצירת עץ השימושיות, המהווה את נקודת הקישור בין המאפיינים האיכותיים המייצגים את נקודות החשיבות של הארכיטקטורות, לבין התרחישים שהוגדרו.
- דוגמה מפורטת לשימוש במתודולוגיית ה-ATAM אפשר למצוא במסמך של אוניברסיטת קרנגי-מלון, המתאר את המתודולוגיה ואת השימוש בה [36].

8.7. שימוש במודל ATAM

כלי ה-ATAM היא הבחינה שבאמצעותה נשווה את שתי הארכיטקטורות, ללא צורך במימוש מלא ויישום של שתיהן:

(1) ארכיטקטורה ל-ESB סטנדרטי מבוסס על Web Services, השכיחים כיום.

(2) ארכיטקטורה ל-ESB המבוססת על ה-Grid.

דרך ההשוואה נבחן את היתרונות ואת החסרונות של כל ארכיטקטורה אל מול סדרת מתארים של תרחישי השימוש שיוגדרו בהמשך. על מנת להשוות בין ארכיטקטורות אלו נגדיר:

- Use Cases, שיתארו את התהליכים הספציפיים במערכת, כאשר כל כמה תרחישים ייצגו אחת או יותר מתשע משפחות השירותים השונים, שמייצגים את יכולות התשתית המוצעת.
- Use Cases, שיתארו את השימוש הלא-פונקציונלי במערכת, עם דגש ליכולות הנדרשות ממערכת תשתית מבוזרת, קרי תרחישי גדילה (Growth Scenarios), ואת העומסים שהמערכת יכולה לעמוד בהם דרך תרחישי העומס (Exploratory Scenarios).
- מציאת מאפיינים איכותיים להשוואה בין הארכיטקטורות – הגדרת נקודות רגישות ונקודות תחלופה (tradeoff) בין מאפייני השימוש בתשתית השירותים המבוזרת.
- בחינה והשוואה של התרחישים, עם המענה של כל ארכיטקטורה לדרישות שהתרחיש מייצג. שלב זה יתבסס על עץ השימושיות, המקשר בין כל תרחיש למאפיינים האיכותיים הנגזרים ממנו. שלב זה יסכם את איכות המענה של כל ארכיטקטורה והתאמתה לדרישות התשתית שאנו מנסים ליישם.

דרך הגדרות אלו נוכל לבחון בצורה מסודרת והשוואתית את התנהגות הארכיטקטורות בתרחישים השונים ואת יתרונות הארכיטקטורה המוצעת.

8.8. מאפיינים איכותיים להשוואה

לאחר בחירת המתודולוגיה להשוואה בין הארכיטקטורות השונות, נקבע סדרה של מאפיינים איכותיים שבאמצעותם ננקד אותן לצורך בחירת הארכיטקטורה העדיפה. עבור כל אחד מן התרחישים ננקד את אופן המימוש על בסיס המאפיינים הבאים:

- מענה מיטבי לתפיסת תשע משפחות השירותים.
- אבטחת מידע – כל התעבורה בין הלקוח ובין השירותים תהיה מוצפנת, כולל בדיקת ההרשאות למשתמש. אבטחת המידע מהווה מאפיין איכותי, כיוון שב-Grid ניתנים שירותים רבים ושונים ללקוחות רבים בעלי מאפיינים מגוונים, ולכן חשוב להפריד ביניהם, לשמור על פרטיותם ולדאוג לאבטחת המידע.
- זמינות השירות – כלל השירותים כוללים את שירותי הגילוי בעלי זמינות של 99.9%. גם אם משאב מסוים אינו זמין, הלקוח יוכל לפנות למשאבים חלופיים.
- גמישות לשינויים – הוספה ופרסום של שירות חדש ו/או של מאפיינים חדשים תהיה מהירה ולא תפגע בזמינות השירותים הקיימים. הוספת המאפיינים החדשים לשירותים, למשאבים ולמוצרים תהיה נוחה.

8.9. הגדרת נקודות רגישות ונקודות Tradeoff

נוציא את הנקודות הרגישות מתוך הגדרת המאפיינים האיכותיים ולאחר מכן נוהה את נקודות ה-Tradeoff :

- אבטחת המידע (Security) :

- i. הצפנה (Encryption) – מרכיב הכרחי להסתרת המידע בסביבה מבוזרת ודינמית.
- ii. הזדהות (Authentication) – מרכיב בסיסי (יסודי) שעליו מתקיימים כל שאר שירותי אבטחת המידע. לדוגמה אפילו לפני ביצוע ההצפנה יש לאמת את זהותו של הצד השני, וזה עוד לפני שמסנכרנים איתו את מפתח ההצפנה.
- iii. הרשאות (Authorization) – כיוון שב-Grid נמצא את אכלוסם של שירותים רבים ברמות הרשאה שונות, חיוני להפריד ביניהם ולהגן עליהם בהרשאות המתאימות.

- זמינות השירות (Availability) :

- i. משאבים חלופיים (Alternative Resources) – חשוב בשל ההתמודדות עם תקלות ומשברים.
- ii. יכולת גידול (Scalability) – חשוב בשל ההתמודדות עם עומסים הנגרמים מגידול בדרישה לשירות.
- iii. אוטונומיה (Autonomy) – היכולת לעבוד לאורך זמן ללא קשר לשירותים האחרים.

- גמישות לשינויים (Agility) :

- i. מהירות התגובה (Performance) – גמישות לשינויים נמדדת כשלוקח למערכת להתאים את עצמה לשינוי שמתרחש.
- ii. דינמיות – קובעת עד כמה המערכת תהיה גמישה. מהם נגזרים ספי הגמישות של המערכת בהוספה, בהורדה ובהתמנות על השירות.
- iii. פשטות (Simplicity) – חשוב לשמור על מימוש הגמישות כדי לא להכביד על השירותים המסופקים (קלות מימוש הגמישות).

יש כמה נקודות חליפיות (Tradeoff Points) בין הנקודות הרגישות שהוגדרו :

- א. הצפנה $\leftrightarrow$ מהירות התגובה - ככל שמעלים את רמת ההצפנה כך פוגעים בביצועים ביחס ליניארי.

- ב. הרשאות $\rightarrow \leftarrow$ דינמיות – ככל שרמת ההרשאות פרטנית יותר ומשתנה עקב שינויים במאפייני המשאב, הדינמיות תיפגע ותהפוך למוגבלת יותר, בשל הצורך להגדיר את ההרשאות באופן פרטני לכל משאב בנפרד.
- ג. משאבים חלופיים $\rightarrow \leftarrow$ פשטות – הצורך לתמוך במשאבים חלופיים לכל שירות פוגעת בפשטות. הצורך בהגדרת האלטרנטיבות, סנכרון המידע, שמירת State עדכני ועוד.

8.10. תרחישי השימוש (Use Cases) במודל ההשוואה

את התרחישים הנ"ל נבסס על מודל "תחנת המוניות", הנפוץ בספרות להשוואת ארכיטקטורות ופתרונות מבוססי Grid (כגון ShanghaiGrid as an information service grid [38,37], שכן המודל מדמה בעיות הקשורות לפעילות אמת המחייבת יעילות במציאות משתנה ודינמית, בעלת מאפיינים ביזוריים, כפי שהוא מאפיין את הארכיטקטורה המבוססת על ה-Grid. בתרחישי השימוש במחקר זה, המודל של "תחנות המוניות" מהווה תרחיש שלם של מרכז שליטה מרובה, המפעיל משאבים דינמיים הזמינים חלקית, במגוון אמצעי קשר, המתמודדים עם תנאים המשתנים בשטח. במקרה זה נגדיר תרחיש שבו "תחנת המוניות" מעסיקה נהגי בית, שאליהם היא מנתבת "נסיעות ספיישל". נוסף לכך, היא עובדת עם נהגים "עצמאיים" ואף עם תחנות מתחרות המשלימות אותה במקרה הצורך. ייתכן שכל אחד מנהגי המוניות יאסוף נסיעות "ברחוב". לכל נהג מונית יש מאפיינים שונים (מספר המושבים, רכב יוקרה, מנשאי מזוודות וכו') ומאפייני איכות שונים שנצברו במערכת (עמידה בזמנים, מהירות הגעה ליעד וציוני בטיחות, כמו מעורבות בתאונות ודו"חות תנועה). התחנה נמדדת בכמה מאפיינים, שהעיקריים מתוכם: רווחיות, % זמן המוניות משולם מהמוניות בתחנה ושביעות רצון הלקוח הנגזר מהשירות, מהתאמת הרכב, מדיוק זמן ההגעה וממהירות ההגעה ליעד. בהנחה שהחברה רשתה את כלל הרכבים באמצעי קשר ובמערכת המחשוב, אשר מסוגלת לספק את התובנות הרצויות, נבחן (בטבלה 6) את התרחישים העומדים בפניה ואשר יוגדרו במסגרת מודל זה:

מספחת השירותים ב- ESB	תרחיש נבחר	תיאור
שירותי אבטחת מידע	הזדהות והרשאות	תהליך ראשוני שבמסגרתו כל אחד מן המשתמשים במערכת, בין שהוא נהג מונית, מנהל תחנה, מזמין מונית, בעל מונית עצמאי או שחקן צד ג' המזדהה עם המערכת. על בסיס זיהוי זה אפשר יהיה להחליט אילו שירותים ומידע לחשוף בעבורו: פעולות הזמינות למנהלי התחנות, הזמנות הפתוחות לחברים בתחנה, לנהגים עצמאיים ולתחנות המתחרות
שירותי שינוע	Notification	תהליך זה מאפשר התמנות על מקור המידע: מאגר ההזמנות

המסרים		לנסיעות וקבלת התראות מידיות ברגע שיש הזמנה שתואמת את יכולות המונית, זמינות הנהג לביצוע הנסיעה ומיקומם.
שירותי אירוח השירותים	טרנזקציות ואישור האירוע	תהליך, אשר במסגרתו הנהג מבצע אישור על קבלת העבודה לאחר ששירות שינוע המסרים התריע על עבודה מתאימה לו ולמונית, בעוד המערכת מבטיחה כי עבודה זו הוקצתה עבורו בלבד ולא עבורו ועבור נהג נוסף בו-זמנית.
שירותי ממשק המשתמש	תצוגה	על ממשק המשתמש של סדרן המוניות להתמודד עם תצוגת נתונים המגיעים מתחנות שונות ומוניותיהן. במקרים מסוימים יש צורך בשיתוף פעולה עם תחנות חדשות, שלא עבדנו איתן בעבר באופן דינמי, ותוך סנכרון תצוגות הנתונים.
שירותי אחסון המידע	ניהול האחסון	שתי תחנות המוניות נמצאות באותה עיר, פועלות בשיתוף פעולה ומגבות זו את זו. אם המערכת קורסת באחת התחנות, הבקרה של התחנה האחרת יכולה להמשיך לעבוד באופן רציף על בסיס המידע המאוחסן בתחנה הראשונה.
שירותי שליטה ובקרה (שוי"ב)	ניטור מצב הרכבים	שירות ניטור בכל מונית המזהה את הקונפיגורציה העדכנית של הרכב, כגון מצב המונית, דלק, רכיבים, כשירות הנהג לנהיגה והפצתו בצורה אמינה, כך שלא יופנו לרכב משימות, כאשר הנהג זקוק למנוחה ו/ או יש לתדלק את הרכב.
שירותי גילוי	גילוי תחנות המוניות	תרחיש שבמסגרתו נהג המונית מגלה ברגע נתון (אפילו לכל נסיעה בנפרד) תחנת מונית בעלת תנאים אטרקטיביים ומתמנה אליה.
שירותי שיתוף	Virtual Organizations	תרחיש שבו קבוצת בעלי מוניות עצמאיים חוברים אד-הוק להקמת תחנת מונית וירטואלית. לחילופין חבירה של בעל מונית עצמאי לתחנת המונית, שהוא אינו עובד תחתה. במסגרת תרחיש זה המערכת נדרשת לאכוף את ההבנות בצורה דינמית ומאובטחת.
שירותי תיווך	המרת הנתונים	התמודדות עם חוסר תאימות מובנה של בעלי מוניות שונים שהצטיידו בחומרה ובתוכנה שונות, ומתן תמיכה בהמרת נתונים בין הפורמטים השונים ויצירת שפה משותפת. במקרה זה נבחן את התרחיש שבו על תחנת מונית אחת להתמודד עם נתוני ניטור המגיעים ממגוון מוניות בפורמטים שונים

טבלה 6 - תרחישי השימוש

9. השוואה ותוצאות

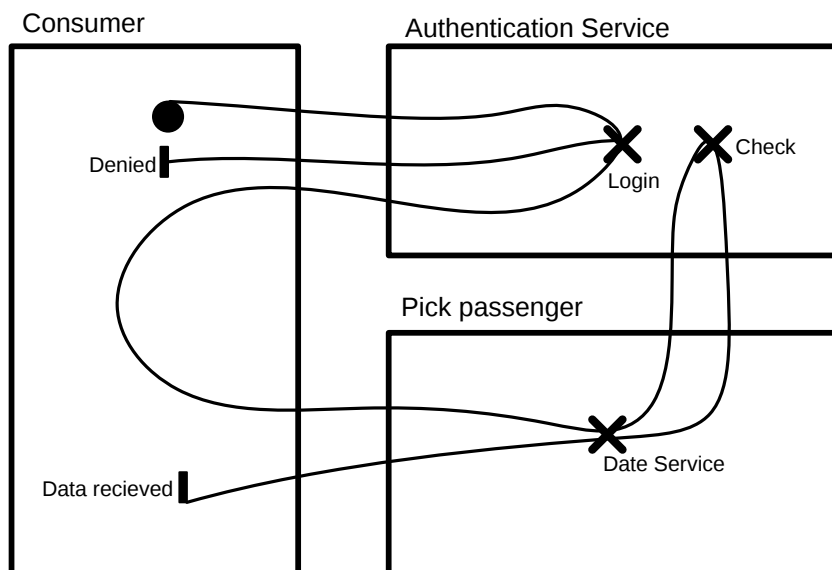
בפרק זה נבחן את תוצאות המימוש, באמצעות מתודולוגית ה-ATAM, שהוצגה בפרק 8, עבור כל אחד מן התרחישים שהוצגו בטבלה 6 בפרק של תרחישי השימוש (8.9), אל מול כל אחת מן הארכיטקטורות: ארכיטקטורה המבוססת על Web Services לעומת ארכיטקטורה המבוססת על Grid Services, כפי שהוצעה בפרקים 6 ו-7. תוצאות המימוש ייבחנו על בסיס המאפיינים שהוגדרו, ונציג תוצאות איכותניות אשר ישמשו אותנו להסקת מסקנות שיוצגו בפרק 10. תחילה, נציג כל אחד מן התרחישים בכל אחד מן המימושים (לפי מודל ATAM) בכל אחת מן הארכיטקטורות, ולאחר מכן, נציג סיכום השוואתי של כלל התרחישים בכל אחת מהן.

9.1. תרחישי ההשוואה

9.1.1. שירותי אבטחת מידע - הזדהות והרשאות

תהליך ההזדהות ומתן ההרשאות עיקרי וחיוני בכל מערכת. בתרחיש תחנת המוניות הוא קריטי מאין כמוהו, כיוון שהצלחתה העסקית מותנית בכך שרק נהגיה המורשים יהיו נגישים להזמנות הנסיעות, וכי סדר החשיפה של ההזמנות לקהל הרחב יותר יישמר. על כל תחנת מוניות, נהג, רכב ולקוח להזדהות מול המערכת, על מנת לפרסם את השירות (הסעת נוסעים) או להשתמש בשירותים הקיימים (התמנות על הזמנות). התהליך הראשוני, שבמסגרתו כל אחד מן המשתמשים במערכת, בין שהוא נהג מונית, מנהל תחנה, מזמין מונית, בעל מונית עצמאי או שחקן צד ג' מזדהה למערכת. על בסיס זיהוי זה אפשר יהיה להחליט אילו שירותים ומידע לחשוף בעבורו: פעולות הזמינות למנהלי התחנות, הזמנות הפתוחות לחברים בתחנה, לנהגים עצמאיים ולתחנות המתחרות.

מימוש באמצעות Web Services

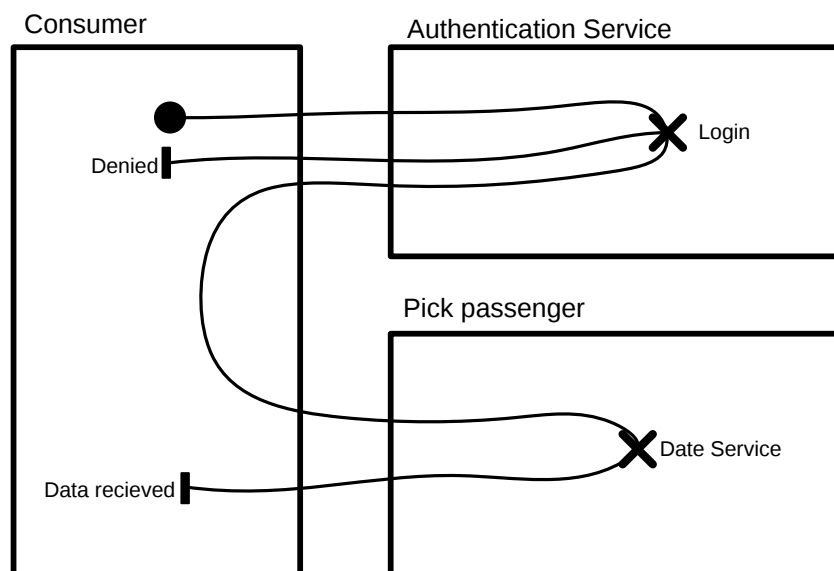


איור 17 - מימוש הזדהות ללא Token

מימוש על בסיס תקני WS-Security, המספקים את פתרון ההזדהות, את ההרשאות, את המדיניות, את הנאמנות (Trust) בין שחקני הקצה ברשת למערכת המרכזית (ובמקרה שלנו בין

נהגי המוניות לתחנות, בין תחנות לתחנות ובין נהגים מתחנה אחת לתחנה אחרת). המימוש נחשב לאיכותי ומספק מענה לצרכי האבטחה של ארגונים מודרניים. הקושי היחיד שבו נתקלת סביבה זו היא מצב של ישויות דינמיות הנדרשות לתקשר ביניהן (שני נהגי מונית) במקרה של נתק מול המרכז (תחנות המוניות).

מימוש באמצעות ה-Grid



איור 18 - מימוש ההזדהות באמצעות Grid token

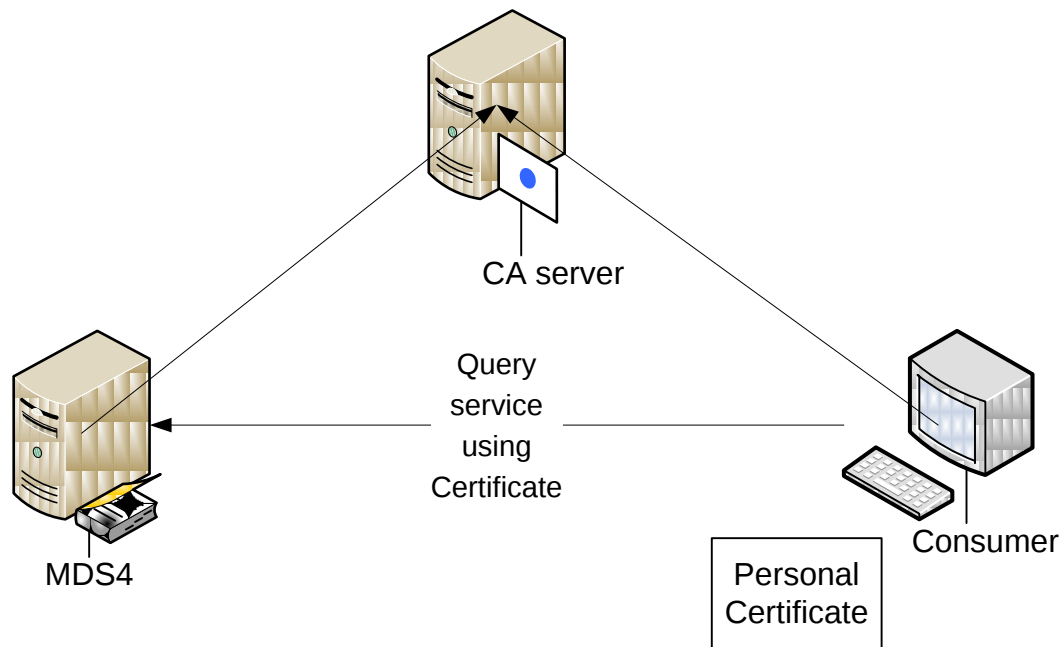
ארכיטקטורת ה-OGSA מתבססת על מפרט ה-WS-Security ומרחיבה אותו, על מנת להשיג אימות (Authentication) של הזהות ואישור (Authorization) לביצוע פעולות שונות. בגישה זו משתמשים בפרוטוקול X.509 (ובפרט ב-X.509 Proxy Certificates) ובסרטיפיקטים (certificates) למשתמשי הקצה. תמיכה בפרוטוקולים אלה מאפשרת לבצע Single Sign-on, כלומר משתמש או משאב מזדהה מול מערכת פנימית ומאותו רגע יכול להזדהות מול כל מערכת אחרת המחוברת ל-Grid (גם מבלי שמפעיל המערכת הגדיר את המשתמש). לדוגמה מנהל התחנה יכול להזדהות פעם אחת ולתקשר מול כל מונית ברשת ישירות או דרך מוניות מתווכות.

נניח שקיום תשתית תקשורתית (הדומה לתשתית ה-IP), המאפשרת לשלוח הודעות למשאב מסוים וקיום נקודת גישה הידועה לו, על מנת שהוא יוכל להתחבר אליה. נוסף לכך, התשתית מאפשרת שידור הודעות מאובטח (כדוגמת SSL/ TLS).

Globus Toolkit מאפשרת שתי אפשרויות לשימוש ב-Certificates:

(1) ה-Certificate מוחזק אצל צרכן השירות (token) וכל פעולות ההצפנה נעשית אצל צרכן השירות. שיטה זו יעילה כאשר לצרכן השירות יש כוח מחשוב המסוגל לבצע פעולות הצפנה.

(2) צרכן השירות מזדהה מול הנציג (proxy) ומזוהה דרכו. שיטה זו מאפשרת לצרכן השירות לבצע את פעולות ההצפנה בצורה עקיפה על-ידי העברת המידע לנציג, שיבצע את הפעולות עבורו. צרכן השירות מזדהה מול הנציג בעזרת סיסמה (כאשר החיבור מאובטח בעזרת SSL לדוגמה), ואז מזדהה מול שירותים אחרים בעזרת תעודה מתאימה, כפי שמודגם באיור 19. לשם כך, GT4 מסופק עם MyProxy – שירות המשמש לאחסון תעודות ולייצוג משתמשים, המאפשר גם להקל על המשתמשים מהצורך לאחסן מידע רב ובעל רגישות במקום לא מאובטח.



איור 19 - שימוש בשירותי הישירות (המשתמש מבצע את פעולות ההצפנה)

שימוש בשיטה זו מאפשר להוריד את התלות בין שירותי ה-Grid המספקים מידע (או פעולות) ובין שירותי ההזדהות וההרשאות ב-Grid, כך שכל שירות בודק הרשאות (וכפועל יוצא, יודע לזהות צרכנים שלא הזדהו ולהפנותם לשירות ההזדהות). שירות ההזדהות מזהה צרכנים ומנפיק להם Token, שבאמצעותו הצרכן יכול לפנות לשירותי המידע ולקבל מהם את מה שהוא זקוק לו באופן דינמי וללא תלות נוספת בשירותים האחרים.

9.1.2. שירותי שינוע המסרים - התמנות לאירוע

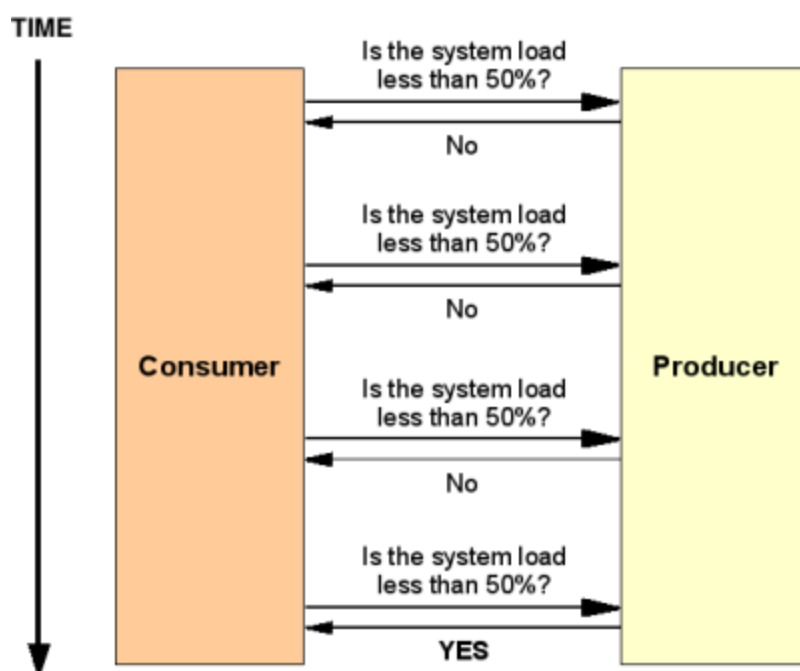
כל אחת מן המוניות במערכת מעוניינת להתמנות לשירות ההודעות, אשר תדווח לה על הזמנות של לקוחות המעוניינים בנסיעה. התמנות על שירות כזה תאפשר לתחנה תגובה מידית וזכייה בעסקאות רווחיות יותר. מימושו של תהליך זה אפשרי לביצוע בשתי דרכים: סנכרונית וא-סנכרונית. מימוש באופן סנכרוני מתבצע בעזרת ביצוע שאילתות על המחשב באופן יזום ומחזורי (לשאול את התחנה כל חמש שניות, האם יש הזמנות רלוונטיות?), כפי שמודגם באיור 20. זוהי דרך בזבזנית, כיוון שנוצר עומס גם על המחשבים וגם על התקשורת, וגם כיוון שהיא אינה מספקת הודעה מידית. דרך נוספת היא בעזרת התמנות לאירוע אצל תחנת המוניות לגבי אירועים מתאימים.

ביצוע בצורה א-סנכרונית מובנה על התמנות הלקוח (לדוגמה נהג המונית) על שירות ההודעות (לדוגמה תחנת המוניות) וקבלת המידע בעזרת דחיפה, מידית, ללא צורך בחידוש הקשר ותקשורת בזבזנית, על מנת לבצע זאת, כפי שמודגם באיור 21. בתגובה, השירות ישמור את הבקשה (הכוללת גם את השירות שאליו יש להודיע על האירוע). כאשר האירוע יתרחש, המחשב ישלח הודעה מתאימה לשירות המצוין בבקשה באופן א-סנכרוני, ואז השירות יוכל להגיב לאירוע.

שיטה זו תומכת גם בצורה פשוטה בשירות Load-Balancing למשאבים על-פי עומס המחשבים המארחים אותם. לדוגמה כאשר בעלי מוניות רבים מתמנים על שירות ההודעות של תחנת המוניות, הרי היא יכולה להעמיד מספר רב של שרתים דומים, אשר ישתמשו בכתובות דינמיות לשרתים. מנגנון חלוקת העומסים יוכל להתמנות על אירועי העומס בכל אחד משרתי תחנת המוניות לבצע איזון עומסים על-פי עומס המחשבים ולשלוח את בעלי המוניות המבקשים להתמנות על השירות לשרת הפחות עמוס.

אפשר לממש את ההתמנות לאירוע, כיוון ש-Grid Services הם Stateful. השירות, המקבל בקשות התמנות לאירועים, שומר אותן אצלו (שינוי ה-state של השירות).

מימוש באמצעות ה-Web Services



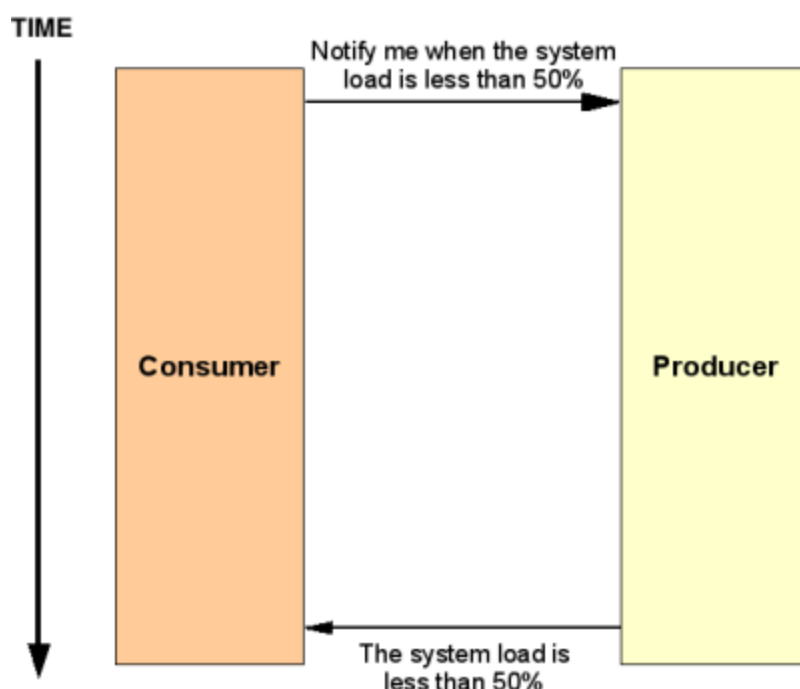
איור 20 - קבלת מידע על אירוע באופן סנכרוני (שליחת שאילתה כל כמה זמן)

תרחיש התמנות זה ממומש על-ידי שימוש במגוון פרוטוקולים מסדרת ה-WS* :

1. זיהוייה של תחנת המוניות מתבצע באמצעות WS-Addressing (יש לציין כי במקרה של משתמשים מבוזרים יש לממש את שירות האיתות, שאינו נתמך ישירות על-ידי סדרת הפרוטוקולים).

2. ההתמנות לרשימת ההזמנות מבוצעת באמצעות שילוב של פרוטוקול WS-Notification, כיוון ש-HTTP אינו פרוטוקול יעיל לניהול ההתראות יש צורך בשימוש ב-WS-MessageDelivery ואילו Efficient Messaging (MOTM) משמש לשיפור יעילות העברת המסרים.

3. אבטחת הגעת ההתראה לנהג המונית מתבצעת באמצעות שימוש ב-WSRM (Reliable Messaging).



איור 21 - התמנות לאירוע וקבלת ההתראה במהלך התרחשות האירוע – מספר הפניות נמוך משמעותית

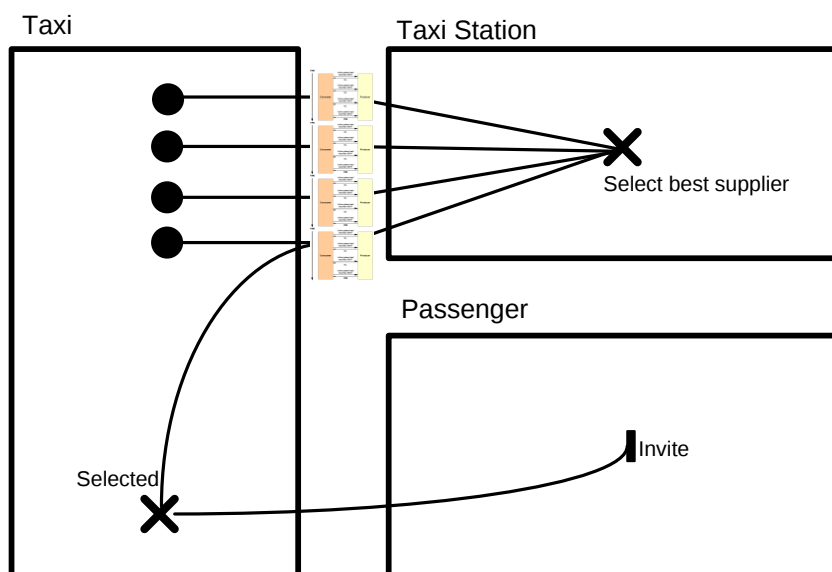
במימוש זה נעשה שימוש בפתרונות הסטנדרטיים של פרוטוקולי ה-WS*. עם זאת, לא נמצא כאן פתרון לבעיה של מציאת תחנת המוניות או התחברות לנהג מונית אחר, המבוצעת על-ידי שימוש בספרייה ושירות הגילוי MDS4. שימוש בספרייה הארגונית MDS4, כפי שהוצג לפני כן:

4. יצרן השירות נרשם באחד משירותי ה-MDS4 הידועים לו.
 5. שירות ה-MDS4 מסונכרן עם שאר שירותי ה-MDS4 המחוברים אליו.
 6. צרכן השירות פונה לשירות ה-MDS4 ומקבל ממנו מידע לגבי מיקום השירות שהוא מחפש.
 7. צרכן השירות מפעיל את השירות אצל יצרן השירות.
- שילוב זה מאפשר תקשורת מיטבית וסטנדרטית תוך התמודדות עם ניידות.

9.1.3. שירותי אירוח השירותים - טרנזקציות ואישור האירוע

תרחיש זה מתאר את אופן אימות הזמנת הנסיעה על-ידי לקוח קצה. במסגרת תרחיש זה, לקוח הקצה פונה לתחנת המוניות ומבקש נסיעה בשעה מסוימת מיעד מסוים ליעד מסוים לשבעה אנשים. הפנייה מוזנת למערכת ומופצת לכלל נהגי המוניות בתחנה. שני נהגים עונים לבקשה, אך הנסיעה מאושרת רק לאחד מהם. לאחר ביצוע האימות ההדדי, מוחזר ללקוח אישור על זמינות הנסיעה, כמו גם לנהג הזוכה.

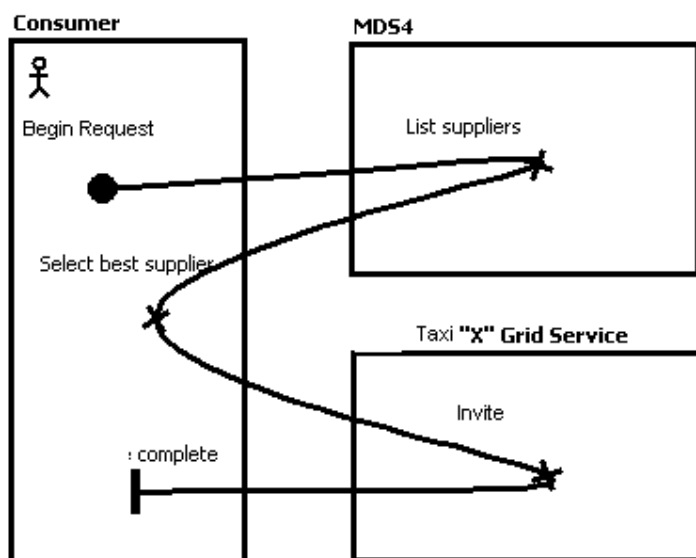
מימוש באמצעות ה-Web Services



איור 22 – מימוש ההזמנה בהתבסס על שירות ההתמנות

לאחר שהלקוחות התמנו לשירות ההזמנות, על מנת לממש תרחיש זה, בתהליך שתואר לפני כן, כמה נהגי מוניות מקבלים התראה על הזמנה אפשרית המתאימה למאפייניהם. כל אחד מהם חוזר לתחנת המוניות על מנת לזכות בהזמנה. ההפנייה מבוצעת על-ידי שימוש ב-Web Service סטנדרטי, כאשר שרתי תחנת המוניות מודאים שרק נהג מונית אחד זוכה בהזמנה באמצעות שימוש ב-WS-TX (Transaction). הפער המשמעותי בתהליך זה הוא הצורך בפיתוח עצמי של לוגיקה לסינון המידע המתקבל מנהגי המוניות לקבלת ההחלטה לאיזה מנהגי המוניות יופץ המידע, ולבסוף סגירת תהליך הטרנזקציה.

מימוש באמצעות ה-Grid



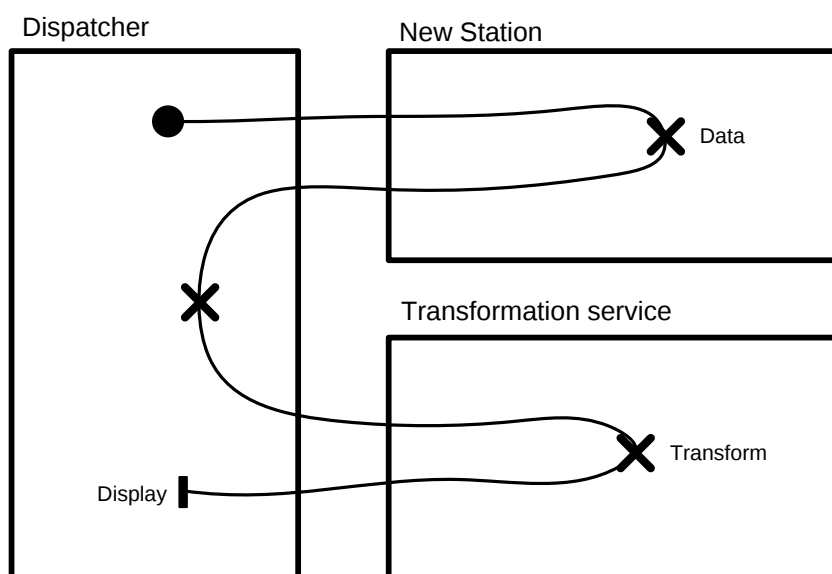
איור 23 – מימוש ההזמנה וביצוע הטרוזקציה ב-Grid

שירותי OGSA פועלים בצורה דומה למתואר בשירותי ה-Web Services. ההבדל המשמעותי הוא השימוש ב-Grid RPC, אשר מאפשר את וידוא אמינות הנתונים המתקבלים מנהגי המונויות, ושירותי Workflow אשר בעזרתם ניתן לבנות תהליך פשוט של מכרז וסגירת זכייה בקלות יחסית.

9.1.4. שירותי ממשק משתמש-תצוגה

בשל ריבוי השירותים במערכת והצרכנים בה, תחנת המונויות (תחנות שונות ונהגים שונים העושים שימוש במגוון חומרות ופרוטוקולים) נדרשת לבסס פרוטוקול אחיד, אשר יציג את הנתונים ממגוון הצרכנים ויצרני המידע (מקורות המידע). כל שירות מפותח באופן שונה על-ידי מפתח שונה, בטכנולוגיה שונה ועבור צרכן שונה. בתרחיש זה, על ממשק המשתמש של סדרן המונויות להתמודד עם תצוגת נתונים המגיעים מתחנות שונות וממונויות של תחנות שונות. במקרים מסוימים יש צורך בשיתוף פעולה עם תחנות חדשות, שלא עבדנו איתן בעבר באופן דינמי, ותוך סנכרון תצוגות הנתונים.

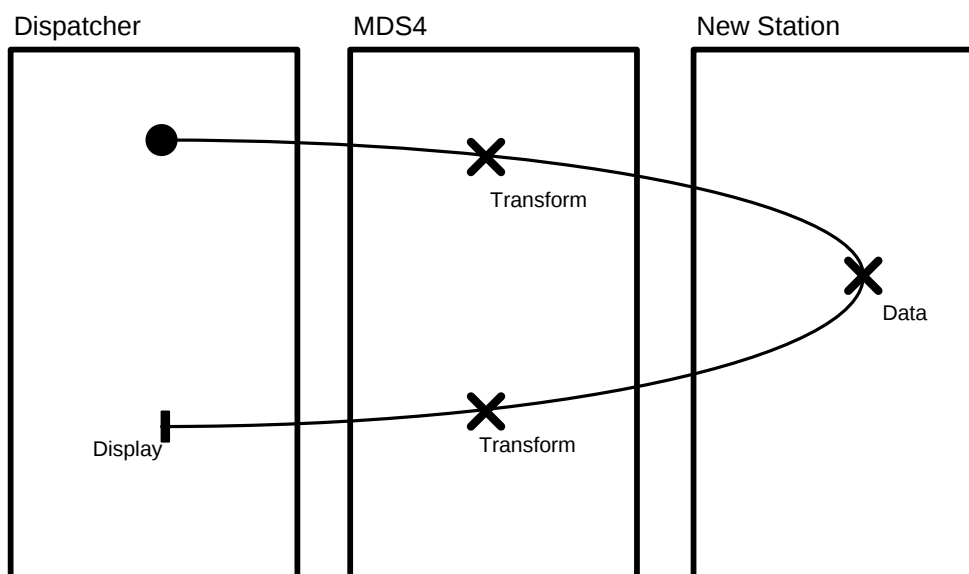
מימוש באמצעות ה-Web Services



איור 24 - מימוש התצוגה באמצעות שירותי המרת נתונים

מענה לחשיפת המידע והצגתו על-ידי גורם שלישי מבוצע באמצעות על-ידי (Web Services Remote Portals). מימוש התצוגה עצמה מבוצע על-ידי סטנדרטים בסביבת הפיתוח, כדוגמת JSR-168 בסביבת Java. כיוון ששירות זה מאפשר רק את ביצוע הקריאה למערכת הרחוקה, נדרש לפתח באופן עצמאי שירותי המרת נתונים שיאפשרו הצגת נתונים מגורמים שונים על גבי ממשק אחיד. המרה זו מחייבת היכרות עם מבני הנתונים המרוחקים.

מימוש באמצעות ה-Grid



איור 25 - מימוש התצוגה ללא תלות במקור המידע

המרת התצורה בתשתיות ה-Grid מבוצעת בצורה דומה למימוש באמצעות ה-Web Services. היתרון בארכיטקטורת ה-Grid היא היכולת להתמודד עם גילוי של השירות בידי הלקוח מהצד השני ללא הכרה מוקדמת ו"התאמת הציפיות מולו" באמצעות שירותי הספרייה, MDS4, המתארים את תכונותיו.

כך, שירות התצוגה יכול לפנות לכל ספק שירות, כולל כזה שמעולם לא פנה אליו קודם ובמסגרת קבלת הנתונים לקבל גם Metadata, שמתאר (ב-XML) את מבנה הנתונים באופן שמאפשר לפנות לשירות מובנה להמרה באופן דינמי ומהיר.

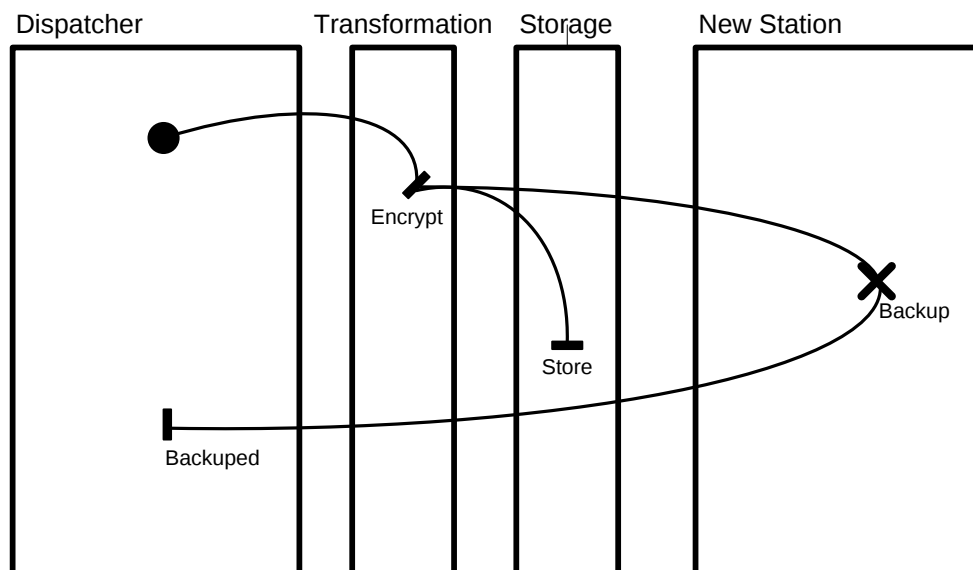
9.1.5. שירותי אחסון המידע - ניהול האחסון

אחסון המידע הוא צורך תשתיתי הקיים בכל מערכת מודרנית. במקרה זה נתמקד בתרחיש של שתי תחנות מוניות הנמצאות באותה עיר והפועלות בשיתוף פעולה ומגבות זו את זו. אם מערכת הבקרה של אחת התחנות קורסת, היא יכולה להמשיך לעבוד באופן רציף על בסיס המידע המאוחסן בתחנה השנייה.

כדי לאפשר את ביצועו של תרחיש זה יש לאסוף את המידע על המוניות הזמינות בארגון, על מנת לדעת אם להפיצו לתחנות השותפות. המערכת נדרשת לבצע את איסוף הרכבים הפעילים והזמינים בכל רגע בארגון, על מנת לדעת אם להפיץ את השדר בתוכו (תחנת המונית) או להתחיל לבזר אותו לתחנות מתחרות ולא לאבד זמן בתהליך הזמנת המונית.

באופן טבעי, רגישות המידע של תחנת המוניות היה מחייב אותה לבחור בקפידה את מיקום הגיבויים. לכן תרחיש זה מחייב התייחסות מיוחדת להצפנת המידע, עקב רגישותו, באופן שיאפשר את אחסונו, למעשה, בכל מקום, אפילו בתחנות המוניות המתחרות.

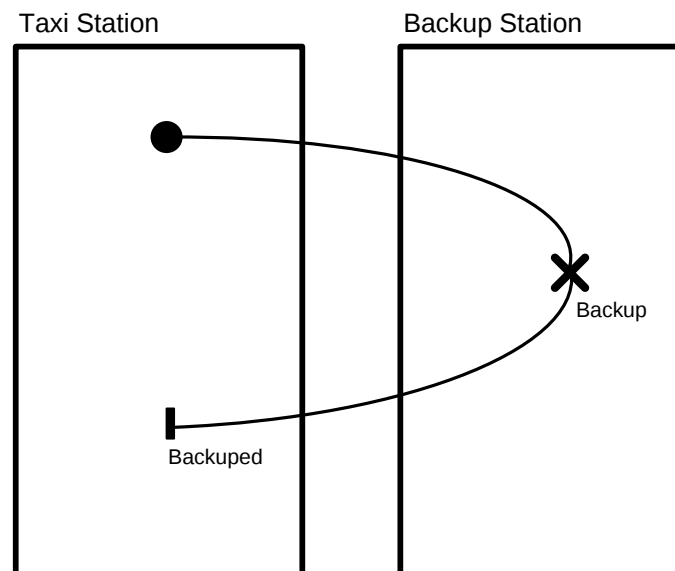
מימוש באמצעות Web Services



איור 26 - מימוש אחסון עצמאי

בתצורה זו אין מענה קיים המאפשר לאחסן בצורה סטנדרטית את הנתונים, למתן מענה לזמינות גבוהה של הנתונים על בסיס Data Replication / או פתרונות Publish/ Subscribe. יש לממש את תהליך איסוף הנתונים ואת שמירתו בצורה זמנית, תכנותית ויזומה.

מימוש באמצעות ה-Grid



איור 27 – אחסון באמצעות MDS4 ו-OGSA DAI

בתשתית ה-OGSA יש מגוון של כלים על מנת לנהל את המידע, כדוגמת OGSA-DAI (Data Access & Integration), הכוללות יכולות של:

- Grid-FTP
- Data Replication
- High-level Publish/ Subscribe
- Encryption

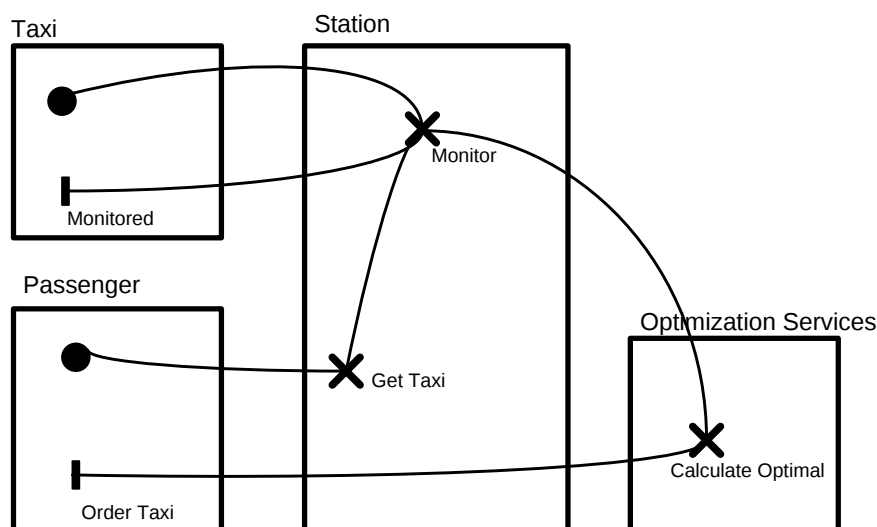
במקרה זה, אנחנו נפעיל את יכולות OGSA-DAI על מנת לאסוף מידע, את יכולות ה-Data Replication, לשמור על מידע זמין בכל נקודת זמן ביותר משרת אחד, ועל-ידי כך נשמור על זמינות מירבית של המערכת גם ברכיבים דינמיים, כגון מחשבי הנהגים, שכל נהג יוכל ליידע את המרכז על זמינותו בכל מקום שבו הוא אוסף את הלקוח. יכולות אלו מאפשרות גם שמירה על זמינות מירבית של יכולות ה-Publish/ Subscribe, המשמשות את נהגי המוניות להתמנות על נסיעות זמינות בתחנת המוניות. יכולת זו מסופקת על-ידי יכולת שירותי שינוע המסרים וה-Notification, שהוצגו לפני כן בתרחיש המתאים.

9.1.6. שירותי שליטה ובקרה (שו"ב) - ניטור מצב הרכבים

מצב המשאבים הוא נתון קריטי בכל מערכת פעילה בשטח הנדרשת לתת מענה בזמן אמת לצרכים המשתנים. ללא ידיעת מצב המשאבים, המשימות יופנו לרכיבים, אשר עמוסים בעומס יתר, בעוד הרכיבים האחרים יוותרו ללא תעסוקה, דבר שיגרום לפגיעה ביכולות המערכת כולה.

בתרחיש תחנת המוניות בחרנו בזיהוי הקונפיגורציה העדכנית של הרכבים, כגון מצב המונית, דלק, רכיבים, כשירות הנהג לנהיגה והפצתו בצורה אמינה, כך שלא יופנו לרכב המשימות, כאשר הנהג זקוק למנוחה ו/ או יש לתדלק את הרכב.

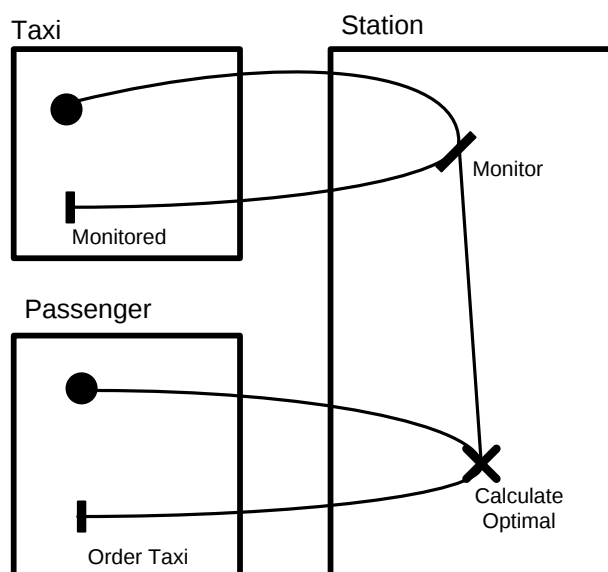
מימוש באמצעות ה-Web Services



איור 28 - מציאת הנהג המיטבי באמצעות נתוני שו"ב

מענה התשתית מבוסס על סדרת פרוטוקולים הנותנים מענה לניהול ולניטור של אמצעי הקצה בצורה המזכירה מעט את תפקיד פרוטוקול ה-SNMP. בין הפרוטוקולים הנ"ל אפשר למנות את: WSDM, WS-Management, WS-Transfer. עם זאת, הם אינם מספקים מענה אמיתי למשאבים הדינמיים, הנדרשים לניהול מלא של מחזור החיים, החל מיצירת המשאב הדינמי (הצטרפות הנהג לתחנה) ועד לסיום תפקידו (פרישתו מהתחנה), כולל ניהול משמרות או יציאה מכשירות.

מימוש באמצעות ה-Grid



איור 29 – מימוש מציאת הנהג המיטבי באמצעות אופטימיזציה

השו"ב

המענה לתרחיש ניטור משאבי המונית מובנה בתוך מודל ה-Grid הקרוי : Grid Economic Model. מודל זה חושף את סדרת השירותים המאפשרים ניהול מלא של מחזור החיים אצל הלקוח במערכת מיצירתו והכנסתו לשירות (ובמקרה זה אימות נהג המונית והמונית והגדרת תכונותיהם). זאת, באמצעות שירותי ההזדהות שהוצגו לפני כן, עבור ניטור מתמיד של יכולותיהם וכשירותם לביצוע הנסיעה וכלה בהוצאתם ממעגל החברים (לדוגמה נידוי עקב נהיגה פרועה). כל היכולות הללו מובנות בתוך שירותי ה-OGSA וכוללות יכולות הקרויות OGSA-based grid workload monitoring, שכוללות את ניטור העומס על כל נהג מונית, זמן התגובה שלו לקריאות. בניגוד למודל ה-Web Services, שבו יש לבצע את מימוש הקוד, אשר ייחשב לכל קריאה את זמני הביצוע שלה, באופן ידני, ה-OGSA כולל תמיכה תשתיתית בניטור אירועים הקורים במהלך הרצת פעילויות, כולל מדידת זמני התגובה, וכך מאפשר מענה מלא לנושא בהשקעה קטנה יחסית. כעת אפשר למפות כל מאפיין של המונית והנהג מבחינה טכנית, שתיתן לי להשתמש באופטימיזציה המובנית ב-OGSA למציאת המונית המיטבית.

9.1.7. שירותי הגילוי לתחנות המוניות

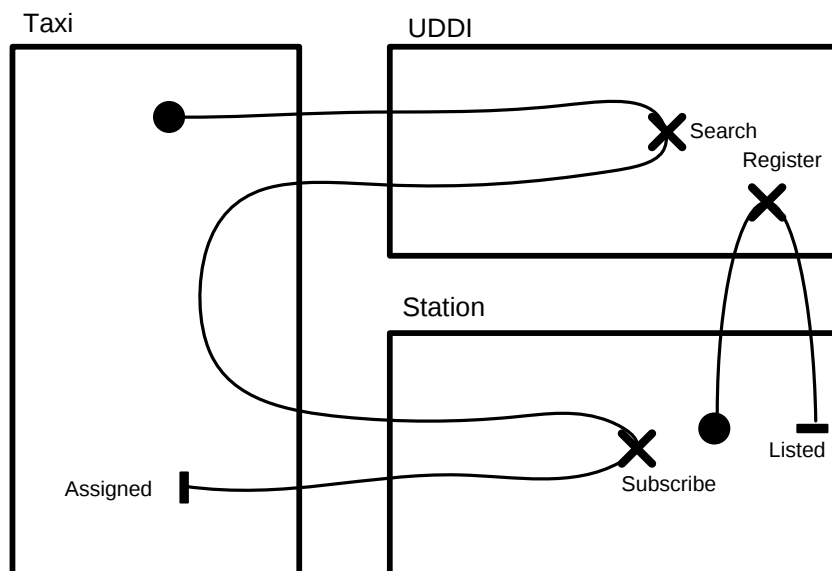
לכל מונית, נהג ומשתמש במערכת צריך להיות זיהוי, על מנת לאפשר לנהג המונית לגלות ברגע נתון (אפילו לכל נסיעה בנפרד) תחנת מוניות בעלת תנאים אטרקטיביים ולהתמנות אליה. מימוש זה דורש שלכל שחקן תהיה כתובת בעולם ה-Grid. לכל משאב, הנמצא על פני ה-Grid יש כתובת, שעל המשתמשים במשאב להכירה. לצרכן השירות יש שתי אפשרויות :

- 1) כתובת סטטית למשאב – הנקבעת במהלך פיתוח צרכן השירות ואינה יכולה להשתנות. היתרון העיקרי של השיטה הוא המהירות של מציאת השירות (כתובתו ידועה מראש).

חוסר הגמישות וחוסר הסקלביליות, לעומת זאת, הם החסרונות העיקריים שלה (אין אפשרות שהשירות יהיה זמין יותר).

2) כתובת דינמית למשאב – לפני הגישה אליו, צרכן השירות מבצע חיפוש כדי למצוא אותו במערכת, שביכולתה להחזיר את כתובת המשאב תוך כדי התייחסות לפרמטרים נוספים (כגון חשיבותו של צרכן השירות, קרבתו הפיזית לשירות וכדומה). כך, אפשר לקבוע מדיניות במערכת ולאכוף אותה בזמן אמת. על מנת לקבל את הכתובת הדינמית, על צרכן השירות להשתמש בשירות החיפוש, אשר ניתן על-ידי המערכת. כאמור, אפשר להשתמש ב-GT4 בשתי הדרכים, על מנת שצרכן השירות יכיר ויוכל להשתמש במשאב מסוים. על מנת להשתמש בשיטה דינמית, מסופק רכיב ה-MDS (Monitoring and Discovery System). זהו שירות, כמו כל שירות אחר, שיכול להיות מותקן על המחשב הצורך אותו (ואז הגישה אליו מקומית ומהירה יותר) או מותקן על מחשב מרוחק.

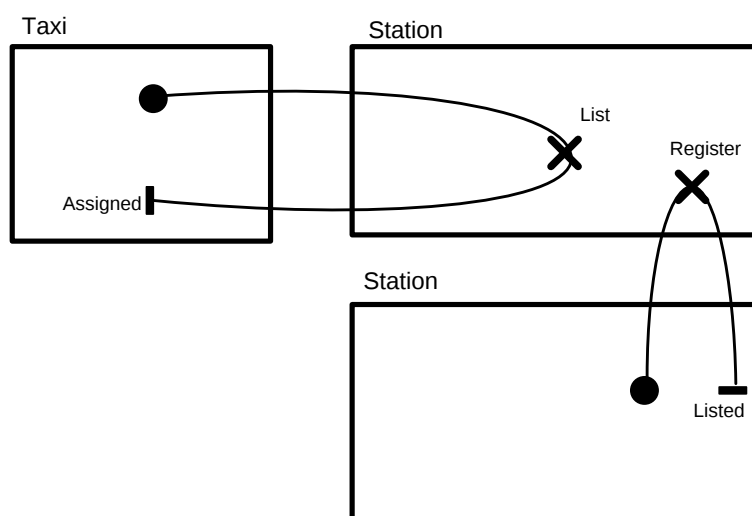
מימוש באמצעות ה-Web Services



איור 30 - תהליך הרישום וגילוי סטטי

בסביבת ה-Web Services גילוי השירותים מבוצע על-ידי סדרת פרוטוקולים סטנדרטיים, כדוגמת UDDI (אשר נמצא בנסיגה בתמיכה שלו בתעשייה), WS-Discovery ו-WS-Resource Framework (WSRF). פרוטוקולים אלה פועלים לפי ההגדרות שפורטו לפני כן, אך בפועל, אף שהם עובדים על-פי מודל הספרייה, שירות ה-UDDI נדרש לכתובת ראשונית קבועה ויש הנחה בסיסית כי יהיה זמין תמיד. ולכן המודל אינו מתמודד בצורה טובה עם סביבות דינמיות, שבהן השרת המרכזי אינו מובטח (ובמקרה שלנו, מצב שבו שני נהגי מוניות חוברים לתחנת מוניות וירטואלית, ללא קיום של תחנה פיזית).

מימוש באמצעות ה-Grid



איור 31 - תהליך גילוי דינמי

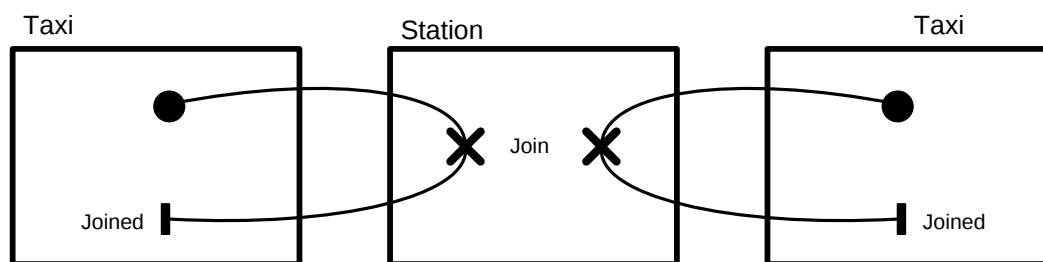
אמנם המימוש בסביבת ה-Grid דומה למימוש בסביבת ה-Web Services, אך השוני המשמעותי ביניהם הוא יכולת הגילוי הדינמי של שירותים ללא תלות בשרת גילוי מרכזי. יכולת זו מבוססת על שירות ה-MDS4, שרץ על כל אחד מן הלקוחות במערכת (ובמקרה שלנו המונית), ועל כן מאפשר את חבירתם ללא צורך בישות מרכזית. הקצאת משאבים דינמית מאפשרת לייצר יחידה אורגנית אופטימלית, המאפשרת את ביצוע המשימה. ה-Grid מקצה למטרת המשימה גם ישות ייעודית לשירותי גילוי ולאתור האוטונומיים, אשר מבחינה ייעודית מטרתם לספק את צרכי היחידה ולתקשר עם שאר שירותי הגילוי והאתור ברשת ולדווח על סטטוס היחידה.

בתרחיש זה, כל תחנה דואגת להירשם אצל התחנות הסמוכות לה ולהעביר את נתוני הרישום של התחנות האחרות. כאשר מונית רוצה לקבל את נתוני התחנות, כל שעליה לעשות הוא לפנות לאחת מהן ולקבל את הרשימה. התרחיש משלב גם אמצעי אבטחת המידע, שיאפשרו לאחסן מידע רגיש, כגון העמלה הנהוגה בכל תחנה, בתחנות הסמוכות וזאת באופן שנתון זה יהיה גלוי למונית בלבד ולא לתחנות המתחרות. היתרון הגדול בצורת מימוש זו הוא בחוסר התלות בשירות יחיד/מרכזי.

9.1.8. שירותי שיתוף - Virtual Organizations

בעולם המודרני נעשה שימוש גובר במודל וירטואלי, שבו קבוצת שותפים בעלי קשר רופף ביניהם (כדוגמת פרילנסרים) חוברים לפרויקט חד-פעמי אחד או יותר. מודל זה, הקרוי "ארגון וירטואלי", מאפשר להם להשיג מטרות משותפות ללא צורך ביצירת הסכמים מורכבים ביניהם. עם זאת, על מנת לאפשר חבירת אד-הוק מעין זו, המאפשרת להקים תחנת מונית וירטואלית, או, לחילופין, חבירה לתחנת המונית, גם אם אתה לא עובד אצלה, דורשת תשתית דינמית וחזקה שתאכוף את ההבנות.

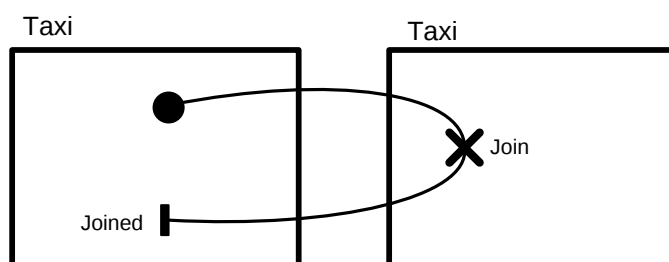
מימוש באמצעות ה-Web Services



איור 32 - מימוש "VO" הדורש רכיב ניהול מרכזי ומוסכם

יכולות ה-Web Services לוקות בחסר בהקשר לנושא זה, כיוון שהוא תואם לארכיטקטורות סדורות יותר ואין לו את היכולות לתמוך במצב שבו אין כלל שרתים מרכזיים. במקרה שלנו, קבוצת נהגי המוניות מחליטים לחבור יחד להקמת תחנת מוניות וירטואלית, ללא משרדים. מימוש "מפעיל וירטואלי" בעולם ה-WS דורש את קיומו של מפעיל שיסכים לספק את כל השירותים הנדרשים לראשון/ לו. לא קיים, למעשה, שירות כזה במימוש ה-Web Services.

מימוש באמצעות ה-Grid



איור 33 - יצירת VO באופן דינמי וללא תלויות

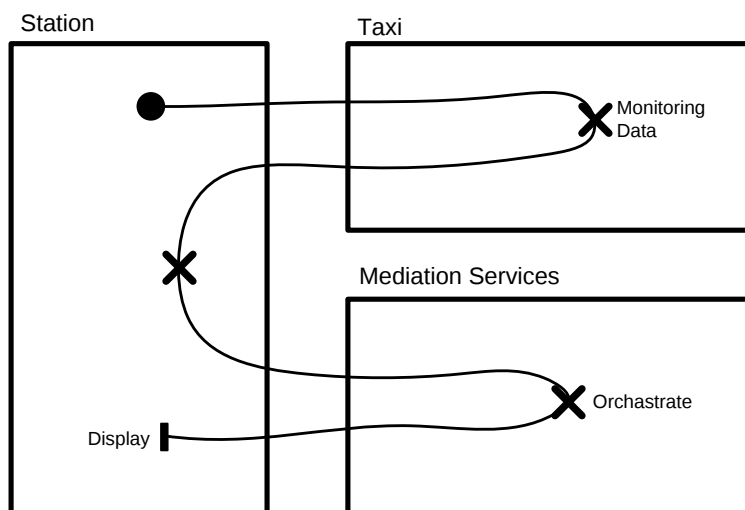
כיוון ששירותי ה-Grid מבוססים על תפיסה של ספריות גילוי דינמיות (MDS), אשר כל אחת מהלקוחות (המוניות) נושאת על עצמה, הרי שאין צורך בשרתים מרכזיים ואין תלות בהם. נוסף לכך, כיוון שההזדהות מבוצעת ונשענת על שירותי תשתית מפתח ציבורי (PKI), הרי שכל עוד התעודה תקינה ושני הצדדים סומכים על מנפיקה, אין צורך בשרת זיהוי מרכזי. **שילוב זה מאפשר מענה מאובטח בצורה נוחה וכולל את מימושם בארגונים הווירטואליים.** בתרחיש שלנו, כל גודל של קבוצת נהגים עצמאיים (משניים ומעלה) יכולים לחבור ולהקים VO, שיינתנו מן התשתית המשותפת שנוצרת.

9.1.9. שירותי תיווך - המרת הנתונים

בעבודה בסביבות הטרוגניות, כדוגמאות של מערכות הפעלה שונות ומכשירים פיזיים שונים (סלולרי, מחשבים ניידים, מחשבים משרדיים) נדרש מענה, אשר יאפשר את המרת הנתונים בין הפורמטים השונים ויצירת שפה משותפת. זהו, למעשה, הכרח בגופים המממשים "ארגונים וירטואליים" וזאת עקב אי-היכולת לאכוף את האחידות בחומרה ובתוכנה.

במקרה זה נבחן את התרחיש שבו על תחנת מוניות אחת להתמודד עם נתוני ניטור המגיעים ממגוון מוניות שונות בפורמטים שונים.

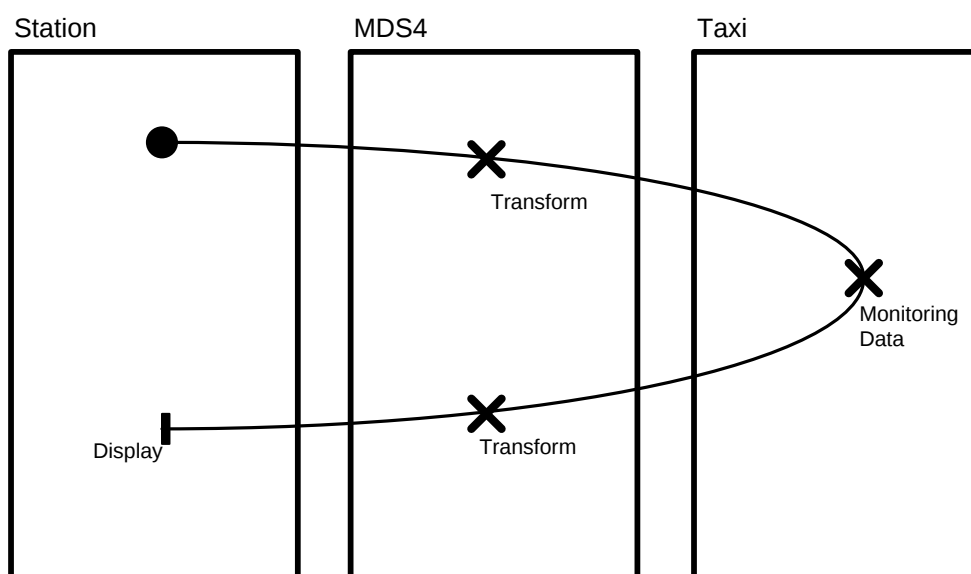
מימוש באמצעות ה-Web Services



איור 34 - המרת הנתונים באמצעות שירות תיווך ייעודי

במסגרת ה-WS* יש סדרה של פרוטוקולים המאפשרים מענה מספק לצורך זה, וזאת באמצעות: BPEL, WS-Choreography, WS-Coordination. שירותים אלה מאפשרים לבצע המרת נתונים, אך מחייבים מימוש ייעודי לכל שירות בנפרד.

מימוש באמצעות ה-Grid



איור 35 - המרת הנתונים באמצעות שירות מובנה

ארכיטקטורת ה-OGSA תומכת באופן מלא ביכולות תמיכה במערכות Legacy וטרנספורמציות המידע, דבר המאפשר חבירה בין המשאבים הדינמיים (הנהגים) לבין מתאם השליטה (תחנת המוניות). מבניות הנתונים (Metadata, שהוזכר לפני כן) מאפשר המרת נתונים תשתיתית כחלק משירותי ה-Grid.

9.2. השוואת התרחישים על בסיס פרמטרים הנבחים ב-ATAM

9.2.1. כללי

ארכיטקטורת ה-Grid מבוססת, למעשה, על שירותים סטנדרטיים, כגון ה-WS*, אשר מטרתה להשלים את החוסרים ולנהל את תהליך הקצאת המשאבים בצורה דינמית ואוטונומית. במרכז הארכיטקטורה נמצאת ישות חשובה, ה-VO, שמאגדת בתוכה הגדרה אחת בעבור יחידה אוטונומית, אשר כוללת את כלל המשאבים והשירותים שהוגדרו בעבור המשימה. יחידה זו משתנה בגודלה על-פי צרכי המשימה. ה-VO מנהל את יחסי הגומלין בין היחידות עצמן דרך ממשק לניהול אחת מהן, אשר מאפשר את שינוי הגדרת המשימה בעבור היחידה ועל-פי התקדמותה. כך אפשר לנהל טוב יותר את העלויות, את הסיכונים, את המידור ואת אבטחת המידע וכמובן לשפר את ביצועי המערכת כולה (זמינות, שרידות ומהירות). גם מרכיבי היחידה משתנים על-פי המשימה ועל-פי התקדמותה. מכך אפשר להבין ש-VO יכול להזמין או לשחרר משאבים על-פי כל שלב במשימה ועל-פי צרכי המשימה המשתנים בכל רגע נתון. ארכיטקטורה סטטית (כזו שאינה כוללת ניהול משאבים והקצאה דינמית/ אופטימלית) מייצרת "רעש" במערכת בעבור כלל הצרכנים המחוברים לערוץ אחד או לצורך ההשוואה ל-VO אחד. במצב כזה, כל המשאבים כבר הוקצו בעבור כלל המשימות ואין אפשרות לבצע אופטימיזציה ואוטונומיה של המשימות השונות שמתרחשות ב-Grid. כך, המשאבים "מבזבזים" ואף מסכנים את הצלחתן של המשימות השונות, שכן הצלחת המשימה תלויה גם בזמינותם של כלל המשאבים בקבוצה. מכאן נובע שתקלה באחד הרכיבים יכולה להשפיע על כלל תפקוד המערכת בצורה מרכזית, כאשר בארכיטקטורת ה-Grid אפשר לנהל את התקלה בעבור קבוצה ספציפית, ובמקרים מסוימים אפילו להשתקם בצורה אוטונומית מן התקלה.

9.2.2. אבטחת המידע

ארכיטקטורת ה-OGSA מייצגת גישה מרחיבה לאבטחת המידע, שבמסגרתה היא דואגת למימוש מלא של כלל רכיבי אבטחת המידע, שעלולים להזדקק להם ב-Grid. שירותים אלה ניתנים באופן עצמאי ובלתי תלוי, הממומשים בצורת שירות. יישום ה-Globus Toolkit דואג לשלב גם את האלמנטים של אבטחת המידע בשאר השירותים באופן תשתיתי, שלא ידרוש מן המשתמש לבצע זאת, ובכך, לא רק שהוא חוסך עבודה, אלא גם מגביר את אבטחת המידע.

לעומת זאת, ESB "קלאסי", המבוסס על תשתיות ה-Web Services, מחייב את מימוש רכיבי אבטחת המידע באופן עצמאי על-ידי תשתיות התוכנה, גם מבחינת השרת וגם מבחינת הלקוח.

9.2.3. זמינות השירות

ארכיטקטורת ה-Grid, המבוססת על ה-OGSA, מספקת מענה לבעיות הזמינות על-ידי תמיכה מובנית בסקלביליות של השירותים – אפשר להוסיף שירותים בצורה דינמית ולשכפלם, להתמנות על אירועי העומסים בשירותים ולהגיב בהתאם. נוסף לכך, שירות ה-MDS הוא Stateful ולכן ביכולתו לעקוב בקלות אחר מצב הזמינות של כל שירותי המוניות ולנתב את הלקוח רק אל השירותים הזמינים.

שירות ה-MDS גם הוא משוכפל ומסונכרן בכמה מופעים ולכן גם הזמינות שלו גבוהה. לעומת זאת, בארכיטקטורת ESB "קלאסית", שירותי גילוי הם נקודת חולשה פוטנציאלית ומהווים צוואר בקבוק במערכת. נוסף לכך, שמירה על זמינות גבוהה של שירות על גבי ה-ESB מצריך תשתיות, כגון (NLB) Network Load Balancing. חיסרון אפשרי של ארכיטקטורה המבוססת על ה-Grid הוא בכך שהשירותים הם Stateful, ולכן נפילת השירות עלולה לגרום לאיבוד המידע שהיה שמור בו.

9.2.4. גמישות לשינויים

הארכיטקטורה נדרשת לתמוך בפרסום ובגילוי השירותים בצורה דינמית. שתי הארכיטקטורות – ESB "קלאסי" ו-Grid תומכים בתקני ה-UDDI ובמתן שירותי הגילוי. ארכיטקטורת ה-Grid בנויה בבסיסה על רעיון ה"משאב" ומאפשרת פרסום וחיפוש משאבים ברשת וגם סנכרון מבוזר של מאגרי הרישום אודות המשאבים השונים בה. כתובת שירות המוניות יכולה להשתנות בקלות ושירותי הגילוי מעודכנים מיד כשזה קורה. בזכות האופי האוטונומי של כל שירות ב-Grid, יש למערכת כולה רמת גמישות גבוהה יותר, כיוון שאפשר להוסיף שירותים חדשים בחופשיות, מבלי להשפיע על השירותים הקיימים.

9.2.5. עץ השימושיות

מהווה את נקודת הקישור בין המאפיינים האיכותיים המייצגים את נקודות החשיבות של הארכיטקטורות לבין התרחישים שהוגדרו.

שימושיות	אבטחת המידע	הצפנה	תרחיש אחסון המידע (9.5), תרחיש הגילוי (9.7)
		הזדהות	תרחיש אבטחת המידע (9.1), תרחיש השיתוף (9.8)
		הרשאות	תרחיש אבטחת המידע (9.1), תרחיש הגילוי (9.7)
	זמינות	משאבים חליפיים	תרחיש שינוע המסרים (9.2), תרחיש אירוח שירותים (9.3), תרחיש ממשק המשתמש (9.4), תרחיש אחסון המידע (9.5), תרחיש הגילוי (9.7)
		יכולת גידול	תרחיש שינוע המסרים (9.2), תרחיש אירוח השירותים (9.3), תרחיש שו"ב (9.6), תרחיש השיתוף (9.8), תרחיש התיווך (9.9)
		אוטונומיה	תרחיש אבטחת המידע (9.1), תרחיש הגילוי (9.7)
	גמישות	מהירות	תרחיש ממשק המשתמש (9.4), תרחיש אחסון

	התגובה	המידע (9.5)
	דינמיות	תרחיש אבטחת המידע (9.1), תרחיש ממשק המשתמש (9.4), תרחיש אחסון המידע (9.5)
	פשטות	תרחיש שינוע המסרים (9.2), תרחיש ממשק המשתמש (9.4), תרחיש השיתוף (9.8), תרחיש התיווך (9.9)

9.3. סיכום ההשוואה

בסעיפים הקודמים נותחו ההבדלים בין מענה ESB על בסיס Web Services ועל בסיס ה-Grid, וזאת על-ידי מתודולוגיית ATAM. על מנת לרכז את הממצאים הרבים באופן ברור, עבור כל אחד מן התרחישים בוצע סיכום ההשוואה בטבלה 7. התוצאות במדרג של 5-1 המסמלים את איכות המענה סוכמו:

1. פער משמעותי – לא קיים מענה ולא ניתן למימוש עצמאי אמיתי.
2. מענה חסר – נדרש מימוש עצמאי מלא.
3. מענה בינוני – נדרש מימוש עצמאי, אך אפשר להסתמך על תשתית קיימת.
4. מענה טוב – יש מענה חלקי שנדרש להשלימו בעזרת הרחבות.
5. מענה מצוין – קיים מענה מלא.

מענה מצוין - 5	מענה טוב - 4	מענה בינוני - 3	מענה חסר - 2	פער משמעותי - 1
----------------	--------------	-----------------	--------------	-----------------

ניתוח התוצאות מראה את היתרון הברור של המענה המבוסס על יכולות ה-Grid, לעומת הפתרונות הקיימים כיום:

מספר השירותים ב-ESB	סטנדרטים רלוונטיים WS*	ציון	סטנדרטים רלוונטיים OGSA	ציון
שירותי אבטחת מידע	מימוש בוגר ומלא מעל לסדרת הפרוטוקולים הסטנדרטיים המצויים בתעשייה יכול להוות בעיה במקרה של עבודה ללא ישות מרכזית בסביבה הטקטית.	4	מענה מלא וכולל לעבודה בסביבה טקטית ללא ישויות מרכזיות.	5
שירותי שינוע המסרים	WS-Addressing, WS-Message Delivery, WSRM (Reliable Messaging), (MOTM) Efficient Messaging, WS-Notification, WS-	4	מתבסס על פתרונות WS* בשילוב עם ספריית MDS4.	5