

**MISKOLCI EGYETEM
GÉPÉSZMÉRNÖKI ÉS INFORMATIKAI KAR**



**ÉPÍTŐELEMES BERENDEZÉSEK
FELHŐALAPÚ SZÁMÍTÓGÉPES TERVEZÉSE
KÜLÖNÖS TEKINTETTEL A SZÁLLÍTÓSZALAGOKRA**

PhD értekezés

Készítette:
Wagner György
okleveles gépészmérnök

Hatvany József Informatikai Tudományok Doktori Iskola
Anyagáramlási rendszerek és logisztikai informatika tématerület
Logisztikai Intézet

DOKTORI ISKOLA VEZETŐ
Prof. Dr. Szigeti Jenő
egyetemi tanár

TÉMATERÜLET VEZETŐ
Prof. Dr. Illés Béla
egyetemi tanár

TÉMAVEZETŐ
Prof. Dr. Illés Béla
egyetemi tanár

TÁRSTÉMAVEZETŐ
Prof. Dr. Kovács László
egyetemi tanár

Miskolc, 2023

Nyilatkozat

Alulírott, Wagner György, kijelentem, hogy ezt a doktori értekezést magam készítettem, és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva átvettem, egyértelműen, a forrás megadásával megjelöltem.

A dolgozat bírálatai, és a védésről készült jegyzőkönyv a későbbiekben a Miskolci Egyetem Gépészmérnöki és Informatikai Karának Dékáni Hivatalában lesznek elérhetőek.

Miskolc, 2023. 09.25

.....

Wagner György

Köszönetnyilvánítás

Az értekezés a Hatvany József Informatikai Tudományok Doktori Iskola keretein belül, a Miskolci Egyetem Logisztikai Intézetében folytatott kutatómunkám eredményeit foglalja össze.

Ezúton szeretném kifejezni köszönetemet és hálámat mindazoknak, akik tanácsaikkal hozzájárultak az értekezés elkészültéhez. Különösképpen szeretnék köszönetet mondani tudományos vezetőmnek, **Dr. Illés Béla** professzor úrnak, szakmai tanácsaiért és projektekbe történő bevonásáért, melyek nélkülözhetetlen segítséget jelentettek a kutatómunkám sikeres elvégzéséhez, a publikációs elvárások teljesítéséhez. Köszönetemet fejezem ki **Dr. Kovács László** professzor úrnak, társtémavezetőmnek, aki számos PhD témavezetése mellé az enyémet is felvállalta, és iránymutatásával, ötleteivel, fejlesztési tanácsaival az értekezés elkészüléséhez nagymértékben járult hozzá.

Köszönöm † **Dr. Tóth Tibor** professzor emeritusnak korábbi kutatási témám témavezetését, a sok konzultációt, és önzetlen segítségét.

Mindig emlékezni fogok, kollégámra, barátomra, † **Dr. Vadász Dénesre**, aki a legösszetettebb kérdéseket is olyan egyszerűen tudta megválaszolni. Tudományos pályafutásomat jelentős részben neki is köszönhetem.

Köszönöm a Gépészmérnöki és Informatikai Kar vezetőségének, valamint **Dr. Szigeti Jenő** professzor úrnak, a Hatvany József Informatikai Tudományok Doktori Iskola vezetőjének, hogy lehetőséget biztosítottak a magas szintű kutatómunka elvégzésére. Hálás vagyok **Homonnai Emesének**, a Gépészmérnöki és Informatikai Kar Dékáni Hivatal dolgozójának, aki a doktori tanulmányaim alatt adminisztrációs feladataimban segítségemre volt.

Köszönöm szeretteimnek, ismerőseimnek a biztatást, akik nálam jobban bíztak ennek az értekezésnek az elkészültében.

Az értekezést

Szüleim emlékének

szeretném ajánlani

*Ha az életben valamibe bele kezdesz,
ne hagyd félbe,
mert életed végéig tartó nyomot hagy benned*

*P. Bozóki István OFM
osztályfőnök
Temesvári Pelbárt Ferences Gimnázium*

Témavezetői ajánlás

Wagner György: „Építőelemes berendezések felhőalapú számítógépes tervezése különös tekintettel a szállítószalagokra” című PhD értekezéséhez

Wagner György a Hatvany József Informatikai Tudományok Doktori Iskola Anyagáramlási rendszerek és logisztikai informatika tématerületéhez kapcsolódóan folytatta PhD tanulmányait. Kutatásai során többek között konvejos és szállítószalagos berendezések tervezésével, működésével foglalkozott és vizsgálta azon kapcsolódási területeket, amelyek mind az informatika mind a logisztika területével kapcsolatosak.

Wagner György diplomamunkáját a Nehézipari Műszaki Egyetem Szállítóberendezések Tanszékén készítette el. Diplomamunkája „Konvejos szállítóberendezések tervezése számítógéppel történő szimulációs vizsgálatával” témakörben készült el, amelyet a tanszék egy K+F projekt keretében vállalt el. A diplomamunka témaköre, kivitelezése és eredményei jelentősen túlmutattak az egyetemi tananyagon. A tanszék javaslatára diplomamunkáját benevezte egy országos diplomamunka pályázatra, amelyen első díjat kapott. Végzése után a Szállítóberendezések tanszéken helyezkedett el. Számos tanszéki kutatómunkában vett részt, amelyek eredményeiről társszerzőkkel publikációk születtek. Az Informatikai Intézetbe való átvétele után a tanszékünkkel nem szakadt meg kapcsolata. Több logisztikai tananyag kifejlesztésében vett részt, és végzett a eredményes kutatómunkát tanszéki projektekben. Informatikai témájú kutatásai során foglalkozott többek között felhőalapú rendszerek kialakításával, működtetésével, biztonsági kérdéseivel.

Wagner György nagy munkabírású, érdeklődő, motivált hallgatóként végezte el PhD tanulmányait. A tanulmányi elvárásokat 1 éven belül, minden tárgy esetében jeles eredménnyel teljesítette, tett sikeres abszolutóriumot, majd

eljárásindítás után summa cum laude eredménnyel doktori szigorlatot. Mind a Logisztikai Intézet, mind az Informatikai Intézet tevékenységébe aktívan bekapcsolódott. Ötleteivel, javaslataival hozzájárult az intézeti kollektívák sikeres kutatási tevékenységéhez. 19 tárgy egyedüli kifejlesztője, egyedüli oktatója, 29 tárgy egyedüli vagy társoktatója. Több mint 100 hallgató szakdolgozat/diplomamunka tervezésvezetője, akik mind sikeresen védték meg szakdolgozatukat/diplomamunkájukat.

Kutatási eredményeit hazai és nemzetközi konferenciákon és folyóiratokban publikálta. Az MTMT adatbázis alapján 69 közleménye jelent meg. 2 könyv társszerzője, 1 könyvrészlet egyedüli szerzője. Független hivatkozásainak száma 11, Q száma 4.011

Az értekezés Wagner György önálló kutatási eredményeit tartalmazza, és minden szempontból megfelel a Hatvany József Informatikai Tudományok Doktori Iskola szabályzatában előírt követelményeknek.

A fentiek alapján a jelölt számára a PhD cím odaítélését messzemenően támogatjuk.

Miskolc, 2023. szeptember 25.

.....

Prof. Dr. Illés Béla
témavezető

Prof. Dr. Kovács László
társtémavezető

Tartalomjegyzék

Nyilatkozat	2
Köszönetnyilvánítás	3
Témavezetői ajánlás	6
Tartalomjegyzék.....	8
Rövidítésjegyzék	10
Használt fogalmak.....	11
1. Bevezetés	13
2. A szakirodalom áttekintése, a kutatás céljai, módszerei	15
2.1 A szakirodalom kutatásának módszertana	15
2.2 A szakirodalom kutatásának eredményei a kutatott évek és a keresési feltételek szerint.....	16
2.3 A szakirodalom kutatásának eredményei a kutatott évek és a publikációs adatbázisok szerint	19
2.4 Az eredmények tudományterületenkénti megoszlása	25
2.5 Következtetések.....	31
2.6 Az értekezés célkitűzései, azokkal kapcsolatos eddigi kutatásaim	32
3. Felhőtechnológiák.....	34
3.1 A felhőtechnológia kialakulása [11].....	34
3.2 Szolgáltatási modellek, felhőarchitektúrák	37
3.2.1 Infrastructura as a Service (IaaS).....	38
3.2.2 Platform as a Service (PaaS).....	38
3.2.3 Software as a Service (SaaS)	38
3.2.4 Felhő architektúrák	39

3.3	Felhőszolgáltatások	41
3.4	Felhőalapú gyártás.....	43
4.	A kidolgozandó rendszer	46
4.1	Modell építőelemes berendezések tervezésére	46
4.2	Felhőalapú építőelemes tervező program architektúrája	49
4.3	A moduláris tervezést végző algoritmus.....	58
4.4	Az algoritmus főbb lépései a következők:	62
4.5	Az algoritmus formális modellje	71
5.	Mintarendszer gumihevederes szállítószalagokra	74
6.	Az értekezés tézisei.....	84
7.	Theses of the dissertation.....	86
8.	Összefoglalás	88
9.	Summary	90
I.	Irodalomjegyzék.....	92
II.	Saját publikációk.....	98
III.	Ábrajegyzék	i
IV.	Táblázatok jegyzéke	iii

Rövidítésjegyzék

API	Application Programming Interface
BIOS	Basic Input/Output System
BLL	Business Logic Layer
DAL	Data Access Layer
DB	Database
DBL	Database Layer
DBMS	Database Management System
IaaS	Infrastructure as a Service
IM	Illesztési Modul
KM	Kereső Modul
MaaS	Manufacturing as a Service
OEM	Original Equipment Manufacturer
OS	Operating System
PaaS	Platform s a Service
PC	Personal Computer
SaaS	Software as a Service
SzM	Szakértői Modul
UI	User Interface
VM	Virtual Machine
VPN	Virtual Private Network

Használt fogalmak

- Gyártórendszer:* egy vagy több berendezésből áll.
- Berendezés:* egy vagy több modulkategóriából (építőelem-kategóriából) áll (például szállítószalag).
- Modulkategória (építőelemkategória):* egy adott feladatot ellátó modul helye (például feszítőhely).
- Modul (építőelem):* egy modulkategória helyre betehető, egy vagy több elemből áll (például görgősor).
- Modultár:* azon modulok összessége, amelyek közül kerülnek kiválasztásra az elvárásoknak megfelelő modulok
- Elemkategória:* egy adott feladatot ellátó elem helye (például görgőhelye egy görgősorban).
- Elem:* egy elemkategória helyre betehető, egy vagy több, az elemnél kisebb darabból áll, amelyek nem elválaszthatók az elemtől (például csapágó).
- Elemtár:* azon elemek összessége, amelyek közül kerülnek kiválasztásra az elvárásoknak megfelelő elemek. Amennyiben nincs ilyen az elemtárban, úgy ebbe az elemtárba fog új elem importálás, vagy kézi adatfelvitel során kerülni.
- Szakértői modul (SzM):* A tervezőtől megkapott adatok alapján meghatározza a berendezés elvárt működését biztosító paramétertípusokat; megkéri a tervezőtől az elvárt működést biztosító értéket; meghatározza a berendezés modulkategóriáit; meghatározza a

modulkategória elvárt működését biztosító paramétertípusokat; meghatározza a modulkategória elvárt működését biztosító határértékeket; meghatározza a modulkategória elemkategoriaiit; meghatározza az elemkategória elvárt működést minimálisan biztosító határértékeket; összeállítja a kiválasztott modulokból a berendezést; ellenőrzi a berendezés elvárt működését.

Kereső modul (KM): Kiválasztja a megadott feltételeknek eleget tevő modulokat; kiválasztja a megadott feltételeknek eleget tevő elemeket.

Illesztési modul (IM): Ellenőrzi a kiválasztott elemek egymással való kompatibilitását, összeépíthetőségét; ellenőrzi a kiválasztott modulok egymással való kompatibilitását, összeépíthetőségét.

1. Bevezetés

Napjainkban egyre inkább terjednek az olyan logisztikai berendezések, amelyek különféle építőelemekből, alkatrészekből épülnek össze. A tervezés ezekben az esetekben tulajdonképpen nem csak a hagyományos, az egyes elemeket érintő mechanikai tervezés, hanem már meglévő építőelemeknek olyan moduláris vizsgálata, hogy hogyan lehet azokból egy kész berendezést létrehozni. Az egyes modulok tervezése mellett feladat a modulok integrálása, illesztése, kompatibilitás vizsgálata. A moduláris tervezés előnye, hogy az összetettebb modulok („építőelemek”) felhasználásával jelentősen felgyorsulhat a tervezés folyamata. A moduláris, logisztikai célú berendezések közül is ki lehet emelni a konvektorokat, a szállítószalagokat, és egyéb más berendezéseket.

A számítógépek, illetve az informatika fejlődése napjainkban lehetővé teszi az előzőleg említett berendezések építőelem elv alapján történő hatékony megtervezését. A tervezés folyamata során a katalógus adatokra építve kell a kívánt berendezés elvárt paramétereit megadni. A rendszer kiválasztja az illeszkedő megoldásokat, amelyekből több is lehet. Az elvárásoknak így több összeállított berendezés is megfelelhet. Ezekből kiválasztva, értékelve, és hangolva adott célfüggvények és optimalizációs módok segítségével választható ki az optimális változat úgy, hogy egy megfelelő célfüggvény segítségével rangsorolhatjuk a megoldásokat.

A korábbi tervezéseknél is lehetséges volt az építőelem elv használata, de ez megkövetelte több nagy terjedelmű, akár 1000 oldalas katalógusok hatékony kezelését. Ezen segít napjainkban az adatkezelés felhőalapú megvalósítása. A számítógépes tervezés felhőalapú megvalósítása miatt egyrészt platform független lesz a tervezést támogató program, így a modulokból építkező tervezés is, másrészt (a lokális telepítésű rendszerekkel összevetve) az adatbázist, és a modulokra épülő

tervező programot is elegendő egy helyen, a felhőben folyamatosan frissíteni, tovább fejleszteni.

A későbbiek során ezt az egész folyamatot szeretném bemutatni egy építőelemes logisztikai berendezésre, egy szállítoszalag tervezésre vonatkozóan.

Összefoglalóan tehát, értekezésemben azt vizsgálom meg, hogyan lehet használni a felhőalapú technika adta lehetőségeket a különböző építőelemekből felépíthető berendezések tervezésénél a hagyományos tervezési eljárások helyett. Példaként a gumihevederes szállítoszalagok tervezését mutatom be.

2. A szakirodalom áttekintése, a kutatás céljai, módszerei

2.1 A szakirodalom kutatásának módszertana

A kutatási terület relevanciáját a témakör publikációs statisztikáján keresztül elemeztem.

A szakirodalom áttekintéséhez olyan korszerű Internet alapú keresőket, keresőmotorokat használtam, amelyek elsősorban a tudományos publikációk főbb adatait tárolják, és lehetőséget biztosítanak különböző szempontok alapján történő kigyűjtésre. Fontos a szakirodalom kutatása során azt is figyelembe venni, hogy a publikációk címében történő keresések sok esetben adhatnak téves találatot. A részletes vizsgálatnál ezért mindenképp egyesével meg kell vizsgálni ezeket a publikációkat, és a vizsgálat eredményét értékelve kell a következtetéseket meghozni.

Témavezetőm javaslatára több keresőmotort is használtam. Ezek:

- a Scopus (<https://www.scopus.com>),
- a Google Scholar (<https://scholar.google.com>),
- a Web of Science (<https://webofscience.com/wos>) valamint
- a Science Direct (<https://www.sciencedirect.com>).

Mivel a Scopus által nyilvántartott cikkek lektoráltak, és minősített folyóiratban megjelent publikációk, ezért a találatok száma itt várhatóan lényegesen kevesebb lehet, mint például a Google Scholar esetében.

A legfontosabb mutató a publikációs intenzitás időbeli változása. Internetes források szerint a felhőtechnikák megjelenése érdemben 2014-re tehető. Bár készültek publikációk a felhőtechnika lehetőségeiről, biztonsági kockázatairól korábban is, de alkalmazni csak ezekben az években kezdték el. Ezért a kereséseket a 2014-2022 évekre fogom korlátozni.

Következő lépés a kereső szavak (kulcsszavak) meghatározása volt. Sok kulcsszó megadása kevés találathoz, míg kevés kulcsszó túl sok találathoz vezethet. Ezért két változatba csoportosítottam a kulcsszavakat, és úgy hajtottam végre több keresést, hogy ugyanazon változaton belül minden egyes keresés eggyel több kulcsszót tartalmazott. Az értekezés címéhez, témájához illeszkedően a következő keresési mintákat használtam:

- (1) - „cloud” AND „planning”;
- (2) - „cloud” AND „logistic” AND „planning”;
- (3) - „cloud” AND „logistic” AND „equipment” AND „planning”.

Illetve:

- (4) - „cloud” AND „modular” AND „planning”;
- (5) - „cloud” AND „modular” AND „logistic” AND „planning”.

Bár várhatóan lesznek átfedések a különböző keresők azonos kulcsszóra épülő keresési találatai között, de ez lényegében nem befolyásolja azt, hogy az elmúlt években született cikkek száma az értekezés témájához arányaiban hogyan illeszkedik.

2.2 A szakirodalom kutatásának eredményei a kutatott évek és a keresési feltételek szerint

A 2022-es év adatainak sorában megjelenő értékek esetében megemlítem, hogy a keresést 2022. júniusában megismételtem annak érdekében, hogy minél frissebb adatokkal tudjak dolgozni. Ezzel együtt is a 2022-es évben megkapott értékek nem az évben megjelenő összes publikációs számot jelentik, hanem arányosan annak várhatóan a felét.

<i>Év/Kulcsszó</i>	<i>(1)</i>	<i>(2)</i>	<i>(3)</i>	<i>(4)</i>	<i>(5)</i>
2014	516	22	5	8	2
2015	499	13	1	3	1
2016	626	8	0	9	0
2017	716	18	2	3	0
2018	786	17	2	10	1
2019	959	19	0	14	0
2020	1048	29	3	12	0
2021	1157	38	3	10	0
2022	556	5	0	6	0

1. táblázat Tudományos publikációk megoszlása
 a **Scopus** esetében [Forrás: Saját szerkesztés]

<i>Év/Kulcsszó</i>	<i>(1)</i>	<i>(2)</i>	<i>(3)</i>	<i>(4)</i>	<i>(5)</i>
2014	58300	2300	4170	3870	2270
2015	61300	2800	4560	4200	2660
2016	68300	3180	5190	5030	3270
2017	76700	4020	5970	5780	3890
2018	75900	5080	7190	6730	5010
2019	79000	6290	8490	7780	5930
2020	77000	7490	10400	8510	7200
2021	59100	9810	12300	9960	8640
2022	21100	5330	6520	5180	4700

2. táblázat Tudományos publikációk megoszlása
 a **Google Scholar** esetében [Forrás: Saját szerkesztés]

<i>Év/Kulcsszó</i>	<i>(1)</i>	<i>(2)</i>	<i>(3)</i>	<i>(4)</i>	<i>(5)</i>
2014	281	2	0	1	0
2015	350	2	0	3	0
2016	387	2	0	1	0
2017	607	15	0	3	0
2018	648	9	2	3	0
2019	846	15	0	6	1
2020	1051	24	1	3	0
2021	1309	29	2	6	1
2022	672	7	1	4	0

3. táblázat Tudományos publikációk megoszlása
 a **Web of Science** esetében [Forrás: Saját szerkesztés]

<i>Év/Kulcsszó</i>	<i>(1)</i>	<i>(2)</i>	<i>(3)</i>	<i>(4)</i>	<i>(5)</i>
2014	4043	289	108	313	34
2015	4939	336	119	403	38
2016	5265	435	145	424	41
2017	5988	486	159	591	63
2018	6630	606	231	618	77
2019	7868	769	259	769	100
2020	9406	1005	347	848	115
2021	12145	1356	510	1154	177
2022	9439	1068	442	910	123

4. táblázat Tudományos publikációk megoszlása
 a **Science Direct** esetében [Forrás: Saját szerkesztés]

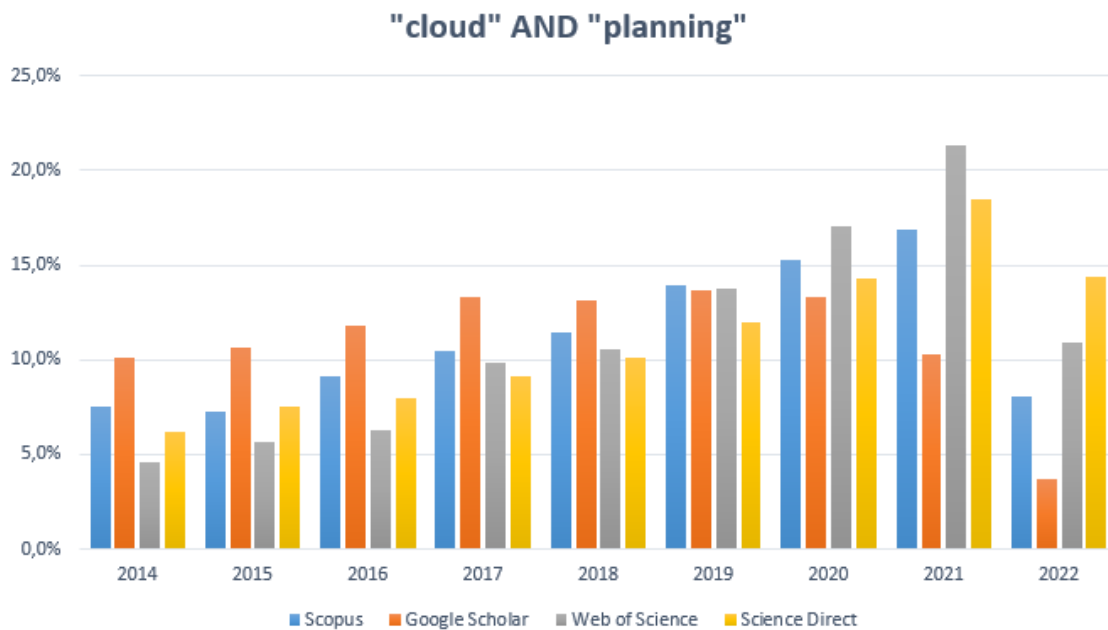
2.3 A szakirodalom kutatásának eredményei a kutatott évek és a publikációs adatbázisok szerint

Ugyanazon kulcsszavak esetén a keresési eredmények a fenti táblázatoknak ebben a szerkezetében nehezen szemléltethetők egy grafikonon. Ezért az eredmények jobb megjelenítése érdekében a keresett kulcsszó variációknak megfelelően 5 újabb táblázatot hoztam létre, amelyeknél a sorok az évszámok, az oszlopok pedig az egyes keresők voltak. Ebben a szerkezetben még jobban láthatóvá vált, hogy ugyanazon keresési feltételek mellett az egyes adatbázisok mennyire eltérő darabszámú publikációt adtak vissza. Ennek csökkentése érdekében minden tudományos adatbázis esetében előállítottam a publikáció a kutatott években megjelent publikációs darabszámok összegét, és annak segítségével százalékokban határoztam meg a kutatott években megjelent publikációk arányát. Ezzel sikerült kiküszöbölni azt, hogy például az első keresési feltétel esetében a Scopus-ból megkapott 516 db-os érték ne vesszen el a Google Scholar-ból megkapott 58300 db-os érték mellett. Az 5. - 9. táblázat ezt tartalmazza, valamint az egyes táblázatokat követően a táblázatokból előállított oszlopdiagramok ezeket az eredményeket szemléltetik.

1. keresés: „cloud” AND „planning”

	Scopus	Google Scholar	Web of Science	Science Direct
2014	8%	10%	5%	6%
2015	7%	11%	6%	8%
2016	9%	12%	6%	8%
2017	10%	13%	10%	9%
2018	11%	13%	11%	10%
2019	14%	14%	14%	12%
2020	15%	13%	17%	14%
2021	17%	10%	21%	18%
2022	8%	4%	11%	14%

5. táblázat Tudományos publikációk darabszámának %-os megoszlása az (1) keresési feltételnek megfelelően [Forrás: Saját szerkesztés]



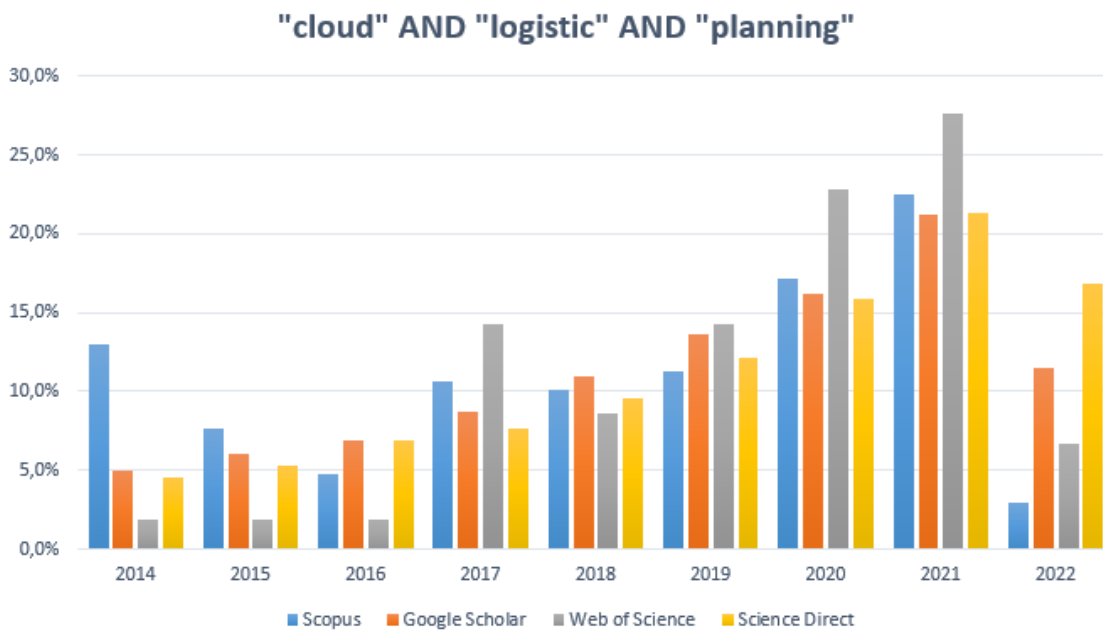
1. ábra Publikációk számának százalékos értéke évek szerint az (1) keresési feltételnek megfelelően [Forrás: Saját szerkesztés]

A Google Scholar kereső eredményei eleinte növekvő számú publicitásra engednek következtetni, amely 2019-ben tetőzött. Ezt követően lassan csökkenő számban jelentek kerültek be az adatbázisába ilyen cikkek. A Scopus, a Web of Science és a Science Direct adatbázisok azonban azt mutatják, hogy a „cloud” AND „planning” kulcsszavakat tartalmazó cikkek száma évről évre különböző mértékben, de növekszik.

2. keresés: „cloud” AND „logistic” AND „planning”

	Scopus	Google Scholar	Web of Science	Science Direct
2014	13%	5%	2%	5%
2015	8%	6%	2%	5%
2016	5%	7%	2%	7%
2017	11%	9%	14%	8%
2018	10%	11%	9%	10%
2019	11%	14%	14%	12%
2020	17%	16%	23%	16%
2021	22%	21%	28%	21%
2022	3%	12%	7%	17%

6. táblázat Tudományos publikációk darabszámának %-os megoszlása
a (2) keresési feltételnek megfelelően [Forrás: Saját szerkesztés]



2. ábra Publikációk számának százalékos értéke évek szerint
a (2) keresési feltételnek megfelelően [Forrás: Saját szerkesztés]

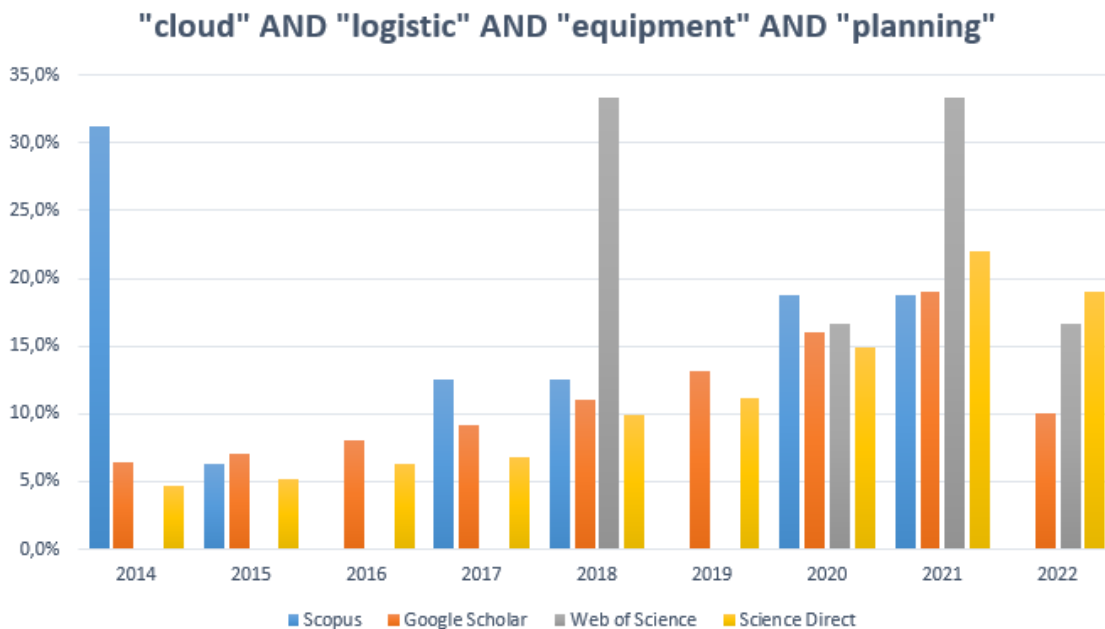
Érdeemes megfigyelni, hogy ha szűkítjük a keresést a „logistic” kulcsszóval,
akkor a Google Scholar az általa korábban is megtalált cikkeken belül növekvő

tendenciára utal, a Scopus csökkenő, Web of Science pedig hullámzó, de jellegében végül is szintén növekvő arányú cikkeket jelez.

3. keresés: „cloud” AND „logistic” AND „equipment” AND „planning”

	Scopus	Google Scholar	Web of Science	Science Direct
2014	31%	6%	0%	5%
2015	6%	7%	0%	5%
2016	0%	8%	0%	6%
2017	13%	9%	0%	7%
2018	13%	11%	33%	10%
2019	0%	13%	0%	11%
2020	19%	16%	17%	15%
2021	19%	19%	33%	22%
2022	0%	10%	17%	19%

7. táblázat Tudományos publikációk darabszámának %-os megoszlása a (3) keresési feltételnek megfelelően [Forrás: Saját szerkesztés]



3. ábra Publikációk számának százalékos értéke évek szerint a (3) keresési feltételnek megfelelően [Forrás: Saját szerkesztés]

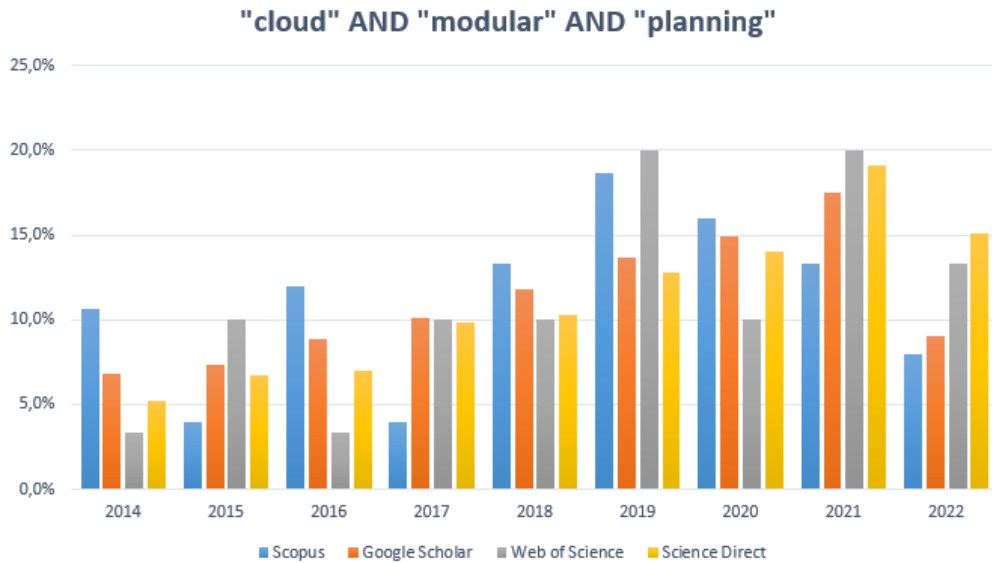
Újabb szóval bővítve az illesztési feltételt, a Web of Science esetében több évben sem került be az adatbázisba a keresési feltételnek megfelelő cikk, és szintén több év esetében a Scopus sem tartalmaz a feltételnek megfelelő cikket, a darabszám 0.

4. keresés: „cloud” AND „modular” AND planning”

	Scopus	Google Scholar	Web of Science	Science Direct
2014	11%	7%	3%	5%
2015	4%	7%	10%	7%
2016	12%	9%	3%	7%
2017	4%	10%	10%	10%
2018	13%	12%	10%	10%
2019	19%	14%	20%	13%
2020	16%	15%	10%	14%
2021	13%	17%	20%	19%
2022	8%	9%	13%	15%

8. táblázat Tudományos publikációk darabszámának %-os megoszlása a (4) keresési feltételnek megfelelően [Forrás: Saját szerkesztés]

A 4-es ábrát elemezve megállapítható, hogy a Science Direct kivételével minden publikációs adatbázis szerint az évek során hullámzó publikációs aktivitás figyelhető meg. Ezt figyelembevéve is jellegében növekvő számú publikációról beszélhetünk a vizsgált időszakban.

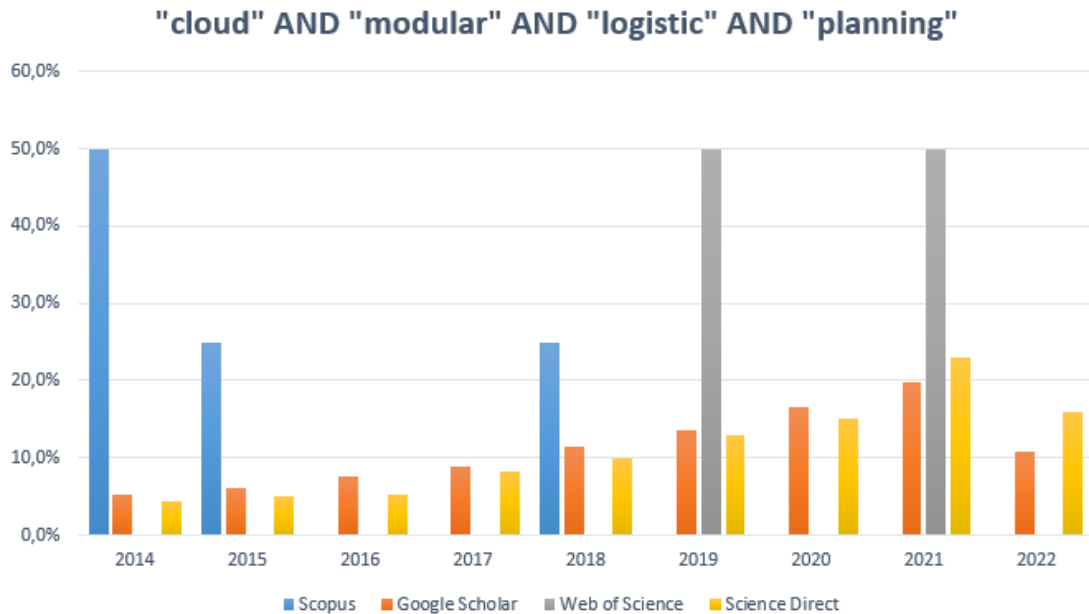


4. ábra Publikációk számának százalékos értéke évek szerint a (4) keresési feltételnek megfelelően [Forrás: Saját szerkesztés]

5. keresés: „cloud” AND „modular” AND „logistic” AND „planning”

	Scopus	Google Scholar	Web of Science	Science Direct
2014	50%	5%	0%	4%
2015	25%	6%	0%	5%
2016	0%	8%	0%	5%
2017	0%	9%	0%	8%
2018	25%	11%	0%	10%
2019	0%	14%	50%	13%
2020	0%	17%	0%	15%
2021	0%	20%	50%	23%
2022	0%	11%	0%	16%

9. táblázat Tudományos publikációk darabszámának %-os megoszlása az (5) keresési feltételnek megfelelően [Forrás: Saját szerkesztés]



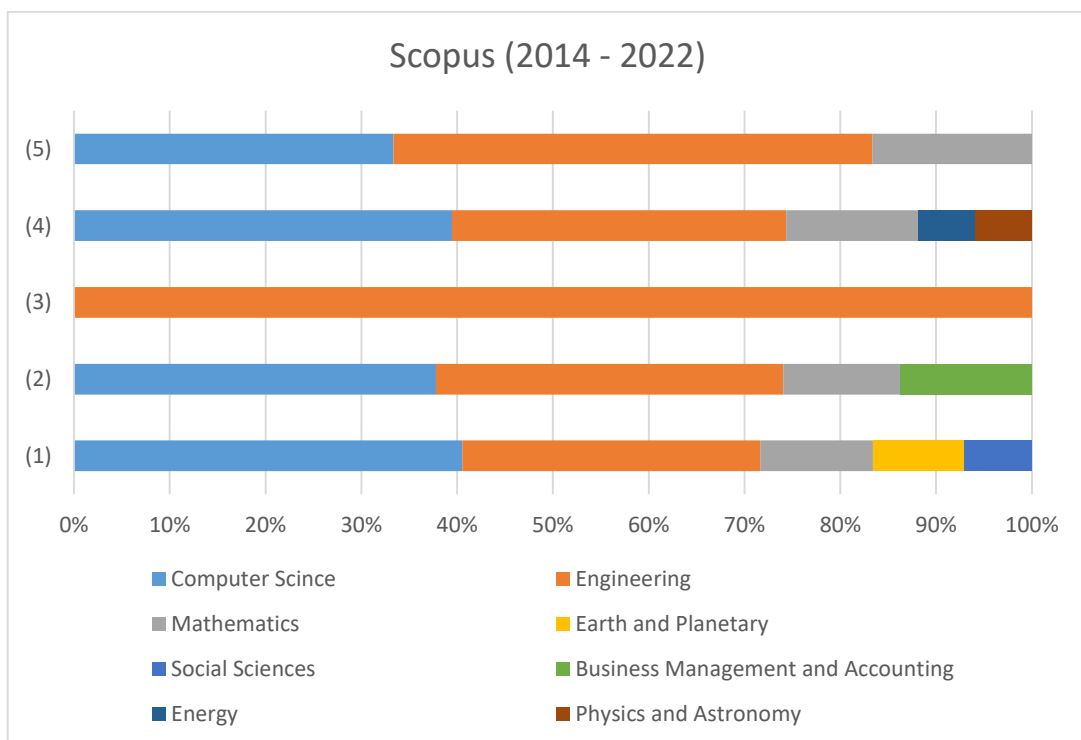
5. ábra Publikációk számának százalékos értéke évek szerint az (5) keresési feltételnek megfelelően [Forrás: Saját szerkesztés]

Mind a Scopus, mind a Web of Science esetében (lásd 9. táblázat) megállapítható, hogy az adott keresési sorozatban az utolsó keresések nem mutattak ki nagyobb számú publikációs tevékenységet. Az egyes években megjelent keresési szempontnak eleget tevő publikációk száma rendkívül alacsony volt, sőt ezek a keresők több év esetében nem is találtak ilyen kulcsszavakkal megjelölt publikációkat. Ennek eredményeképpen, ha a fennmaradó évek valamelyikében készült is 2-3 publikáció, annak darabszáma százalékosan arányítva a többihez teljesen félrevezető következtetésekhez vezethet. A Google Scholar valamint a Science Direct adatbázisaiból kinyert darabszám ugyanakkor egyenletesen növekvő arányú publikációs tevékenységre utal.

2.4 Az eredmények tudományterületenkénti megoszlása

A publikációs adatbázisokból kinyert eredmények helyes értelmezéséhez mindenképpen figyelembe kell venni, hogy a publikáció milyen tudományterületen született. Bár a keresésnél megadott kulcsszavak látszólag jól leszűkítik

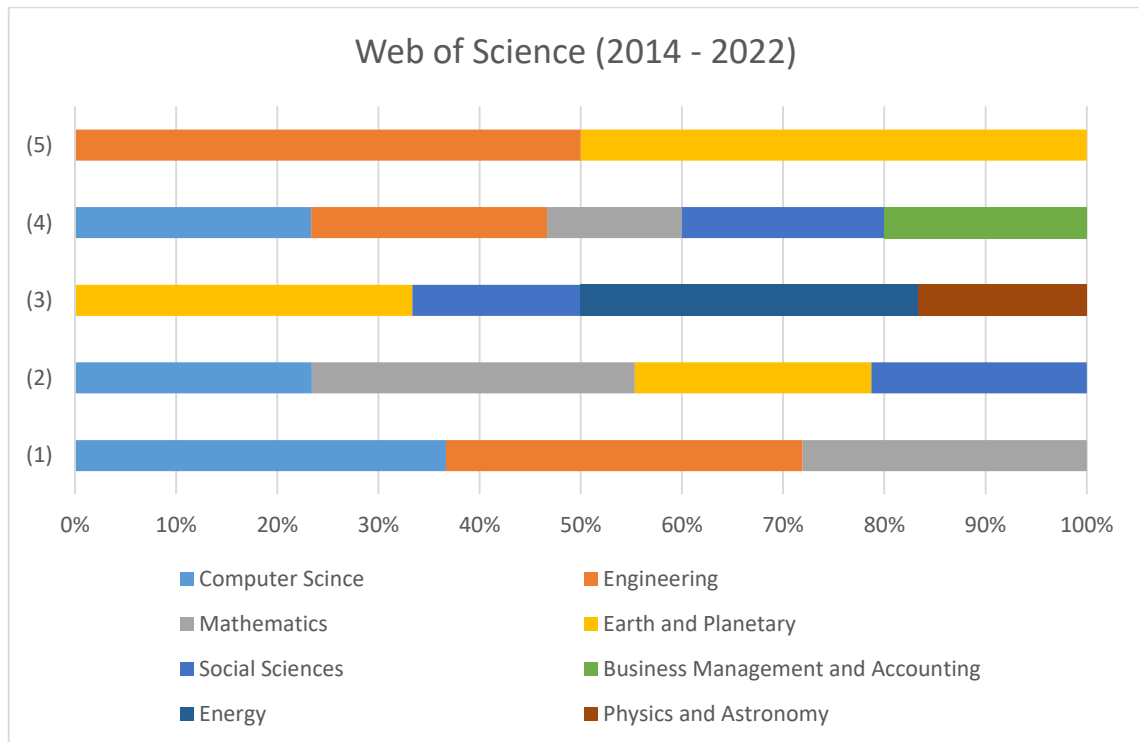
a publikációkat, de a megkapott publikációk áttekintésekor látható, hogy olyan teljesen eltérő tudományterületen készült cikkek is belekerültek a keresés eredményei közé, mint például az „Earth and Planetary”. A publikációs adatbázisok nem azonos tudományterületi elnevezéseket használnak, bár vannak átfedések. Ezért a helyes szemléltetéshez az egyes adatbázisokból megkapott publikációs darabszámokat adatbázisonként külön-külön fogom megjeleníteni, használva a korábban bemutatott, a kutatott években megjelent az összes darabszám segítségével normalizált százalékos értékeket előállító módszert.



6. ábra Publikációk által lefedett tudományterületek megoszlása a **Scopus** adatbázisa szerint [Forrás: Saját szerkesztés]

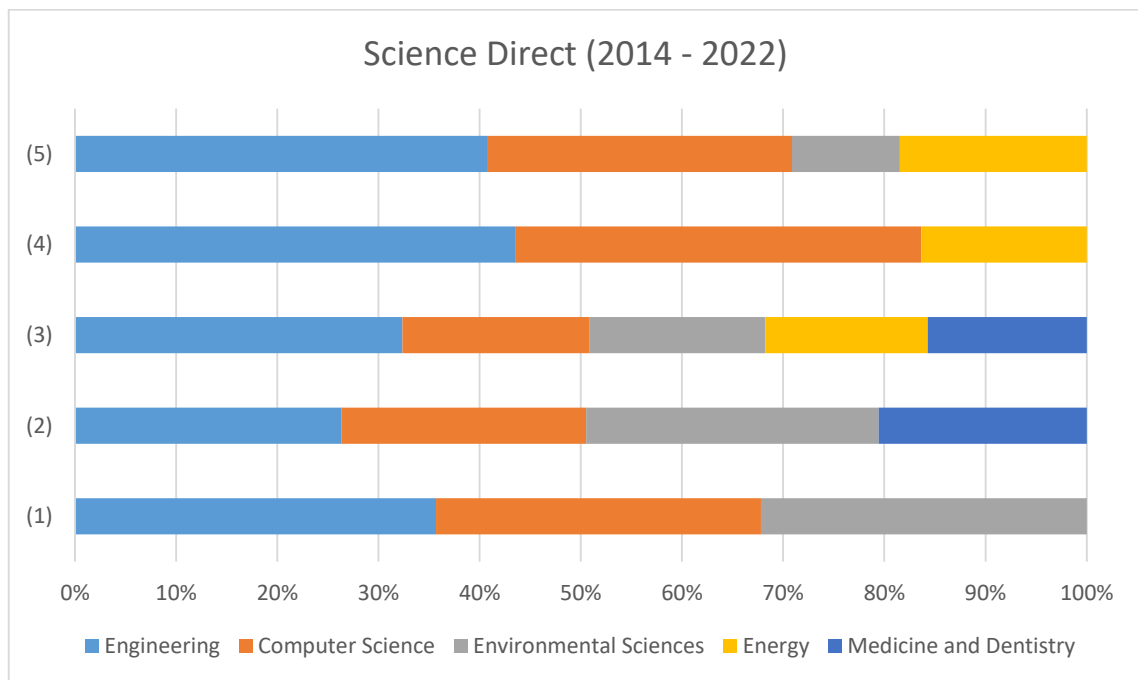
A 6. ábrán egyértelműen látható, hogy az „Engineering” tudományos terület mind az öt kulcsszavas keresés esetében tartalmazott publikációkat, sőt a (3)-as, „cloud” AND „logistic” AND „equipment” AND „planning” keresés esetében csak „Engineering” területen születtek publikációk. Ugyancsak sok publikáció került ki a „Computer Science” tudományterületről. Az öt keresési feltétel közül négy esetében közel 40% körüli arányban.

A Google Scholar adatbázis esetében nem találtam olyan információt, amely tudományterületekre besorolva mutatta volna az egyes keresések esetében a publikációk megoszlását, ezért a Google Scholar-ra épülő keresések esetében a tudományterületenkénti publikációs darabszámokat nem tudom értékelni.



7. ábra Publikációk által lefedett tudományterületek megoszlása a **Web of Science** adatbázisa szerint [Forrás: Saját szerkesztés]

A 7. ábrát tanulmányozva jól látható, hogy az „Engineering” tudományterület magas arányban képviselteti magát az (5), „cloud” AND „modular” AND „logistic” AND „planning” kulcsszavak alapján történő keresés esetében. A „Computer Science” tudományterület három keresés ((1), (2) és (4)) esetében is érintett, 37%, 23% és 23% -os értékekkel.



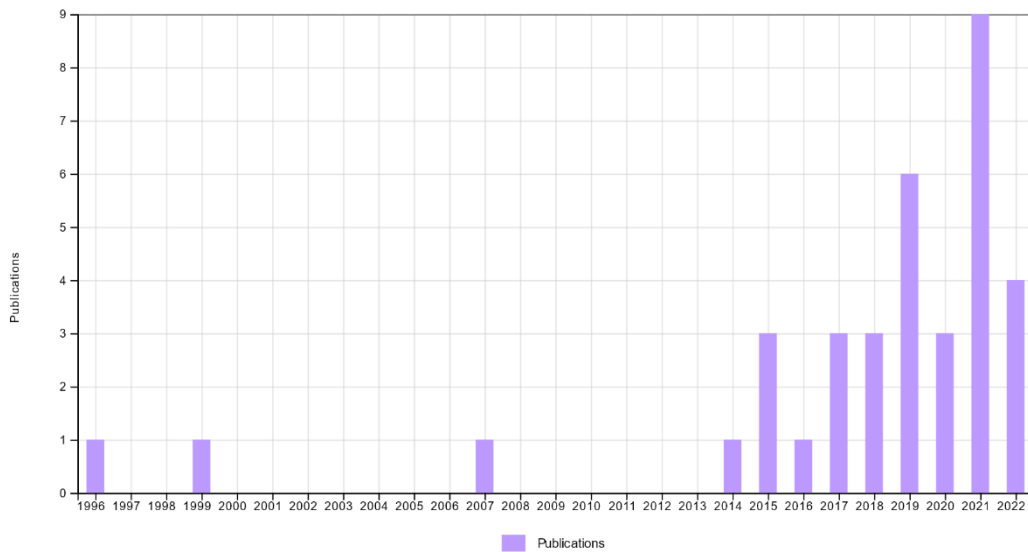
8. ábra Publikációk által lefedett tudományterületek megoszlása
a **Science Direct** adatbázisa szerint [Forrás: Saját szerkesztés]

A 4. táblázatban már látható volt, hogy a Google Scholar és a Science Direct adatbázisok voltak azok, amelyek mind az öt keresés esetében minden évben találtak megfelelő a megadott kulcsszavaknak megfelelő publikációkat. A Science Direct -ből származó eredményeket vizsgálva az „Engineering” tudományterület a kutatott időszakban 26% és 44% közötti arányban tartalmazott publikációkat, hasonlóan a „Computer Science” tudományterülethez (19% - 40%).

Összefoglalóan megállapítható, hogy:

- a kulcsszavak megfelelő szelekciós erővel rendelkeznek, és relevánsak. Nem volt olyan kulcsszó variációra épülő keresés, amely több publikációs adatbázis esetén is üres halmazt eredményezett volna;
- a 9. ábrán látható, hogy a publikáció alapjául szolgáló évek (2014-2022) közel teljes mértékben lefedik a kapcsolódó kutatási periódusú aktivitást;

- a publikációk számának százalékos értéke évek szerint elemezve a legtöbb publikációs adatbázis szerint növekvő számú publikációt jelez, azaz az értekezéshez köthető kulcsszavak alapján a kutatási téma egyre nagyobb érdeklődésre utal;
- a publikációk által lefedett tudományterületek érintik mind az „Engineering”, mind a „Computer Science” területeket, amelyekhez ez az értekezés is kapcsolódik.

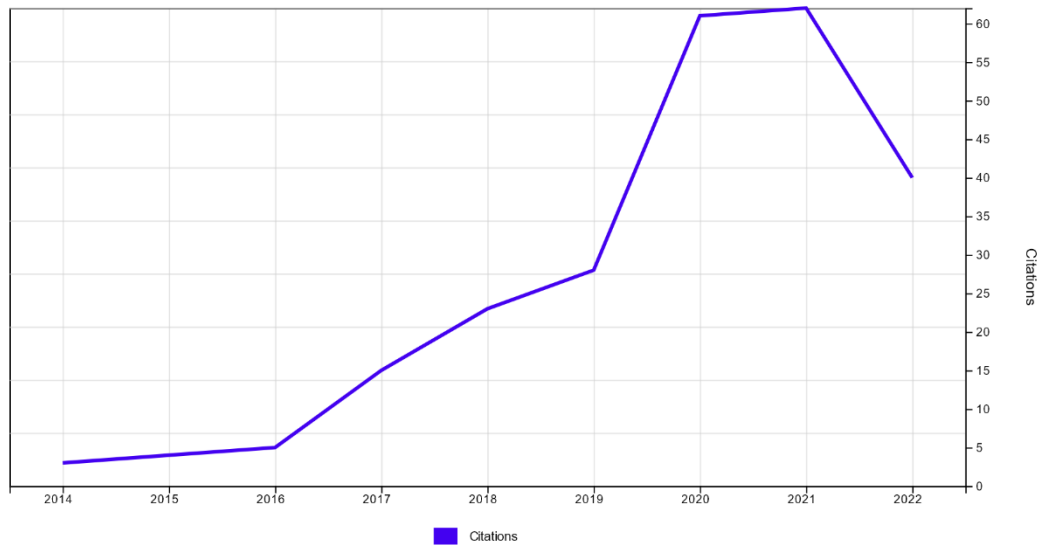


9. ábra Publikációk darabszámának alakulása

a Web of Science adatbázisa alapján [Forrás: Saját szerkesztés]

A vizsgált időszakban, ha a tématerületet lekorlátoztam az „Engineering”, valamint a „Computer Science” tudományterületekre, csak két publikáció maradt benne a szűrőben. Annak érdekében, hogy kiszélesítsem a keresést a lehetséges publikációk számának növelése érdekében, a (4) keresést ismételt meg a 2014-2022 évekre szűkítve, és a fenti két tudományterületre korlátozva. Ezek után 33 publikáció maradt a szűrőben. Érdeemes megvizsgálni, hogy ezekre a publikációkra milyen számban, és éveket tekintve milyen eloszlásban történik hivatkozás.

Ezt szintén a Web of Science alapján tettem meg, a fenti megkötéseket alkalmazva. A hivatkozások számának alakulása a 10. ábrán látható. Egyértelműen látszik, hogy nem csak a publikációk száma, de az azokra történő hivatkozások száma is nő. A diagram utolsó értéke a 2022-es, most még tört évet mutatja. Az ott leolvasott értéket időarányosan vizsgálva az eddigi legmagasabb citációs érték adódik.



10. ábra Hivatkozások számának alakulása
a Web of Science adatbázisa alapján [Forrás: Saját szerkesztés]

Önhivatkozások nélkül 239 hivatkozás található a 33 publikációra. A 10 leg többször hivatkozott publikáció és rájuk való hivatkozások száma az elmúlt 5 évben a Web of Science alapján:

Év / Publikáció	2018	2019	2020	2021	2022	Σ
[1]	8	9	7	6	7	45
[2]	2	4	11	9	1	27
[3]	2	4	9	6	3	24
[4]	4	3	1	1	0	21
[5]	0	0	7	9	4	20
[6]	4	2	6	2	0	14
[7]	0	0	2	5	3	10
[8]	3	2	3	1	0	10
[9]	0	0	2	4	2	8
[10]	0	0	2	5	1	8

10. táblázat A kiválasztott tudományterületeken készült publikációkra való hivatkozások száma a Web of Science alapján [Forrás: Saját szerkesztés]

2.5 Következtetések

A legtöbb hivatkozást kapott publikációkat elemezve megállapítható, hogy a megadott szűrési feltételeket használva ezen publikációknak nagy része csak részlegesen kapcsolódik a kutatási témához. A részlegesen illeszkedő publikációk több esetben gyártórendszereket vizsgálva mutatnak be olyan módszereket, amelyek segítségével a gyártórendszer egyes elemeinek tervezése egyszerűsödik, illetve olyan üzemeltetési algoritmusokat, amelyekkel egy rendszer működési hatékonysága nő. A legtöbbet hivatkozott publikációk között nincs olyan, amely teljesen lefedné az értekezés kutatási témáját. Nem leszűkítve a publikációs kutatást a legtöbbet citált publikációkra, már jelennek meg olyan publikációk, amelyek jobban illeszkednek az értekezés kutatási témájához. A „cloud” és a „planning” illetve a „cloud” és a „modular” kulcsszavak

használatakor megjelenő publikációk esetében jelentős számban fordulnak elő gyártás hatékonyság növelésével foglalkozó publikációk. A gyártórendszerek egyik fontos építőeleme a gyártórendszerek kiszolgálását végző logisztikai berendezés, mint például egy szállítószalag. Egy hatékonyabb, nagyobb volumenű gyártásra képes gyártórendszer hatékonyabb logisztikai berendezéseket igényel. A szállítószalagok tervezése során jól alkalmazható a moduláris tervezés. Egy szállítószalag tervezése jól megvalósítható felhő alapokon, építőelemeken alapuló tervezéssel. A releváns szakirodalom vizsgálata után megállapítható, hogy a publikációk ezt a területet nem kutatták, így a témában jelentős kutatási lehetőség rejlik. Az eddigiek alapján megalapozottnak tartom az értekezés kutatási témáját.

2.6 Az értekezés célkitűzései, azokkal kapcsolatos eddigi kutatásaim

Az értekezés témáját eddigi kutatásaim, azok eredményei és a projektek keretén belül készült publikációim indukálták. Az irodalom elemzése alapján az értekezésnek az alábbiak a célkitűzései:

- ki kell dolgozni egy olyan modellt, amely logisztikai berendezések moduláris (építőelem elvű) tervezését támogatja;
- ki kell dolgozni egy olyan felhőtechnológián alapuló architektúrát, amely alkalmas a fenti modell működésének biztosítására;
- ki kell dolgozni egy olyan algoritmust, amellyel optimalizálható a modulok kiválasztása, integrálása, karbantartása.

A logisztikai rendszerek rendkívül összetettek. Ezek problémamentességét úgy lehet garantálni, hogy az egész rendszert egységében kell kezelni. Ekkor már a tervezés során felmerül az a kérdés, hogy az egyes modulok milyen feladatokat lássanak el, és hogy fognak tudni egymással kommunikálni. Ezért tartom kiemelten fontos feladatnak a logisztikai rendszerek egységes architektúrájának kérdését már a tervezés kezdetétől fogva. Ennek keretén belül foglalkozni kell azzal, hogy az együttműködő rendszerek milyen információkat osszanak meg

egymással. Ezért ki kell dolgozni egy új architektúra modellt. Három TÁMOP projekt keretén belül foglalkoztam ezzel a kérdéskörrel, és készítettem ezekből publikációkat. [S9-S11]

Egyetemi oktatási, kutatási tevékenységemet 1985-ben az akkori Szállítóberendezések Tanszékén kezdtem. Több K+F munkába kerültem bevonásra, ahol munkatársaimmal a szállítóberendezések számítógéppel történő tervezésének optimalizálási kérdéseit, a tervezés elméleti kérdéseit, és azok lehetséges válaszait kutattam, majd készítettem rá különböző számítógépes programokat, amelyek terjedelmük és összetettségük miatt szinte minden esetben több modulban kerültek megvalósításra. [S12-S16]

Mindig fontosnak tartottam az informatikai biztonságot, így a 2013-ban megrendezett OGÉT XXI. konferenciára a felhőalapú tárolás biztonsági kérdéseivel foglalkozó cikket írtam, és tartottam erről előadást. Ugyanezzel a témakörrel foglalkozó cikkem jelent meg a GÉP folyóirat 64-es számában. A felhőtechnológiával később is foglalkoztam, így az értekezés kutatási témájában megfogalmazásra kerülő felhőalapú számítógépes tervezést kifejezetten érdekesnek, és szakmailag is indokoltnak tartom. [S7][S8]

A kidolgozott modellhez implementációs mintarendszert készítettem, amellyel jól demonstrálja a kidolgozott modell helyességét.

3. Felhőtechnológiák

A felhőtechnológiát használó alkalmazások számos előnnyel rendelkeznek. A felhőszolgáltatások Internet elérés esetén bárholnan használhatók. Az alkalmazott technológiától függően maga az alkalmazás futhat a felhőben, és a szükséges adatbázisok is tárolódhatnak ott. A végfelhasználó számítógépe gyakorlatilag kliensként fog viselkedni. A felhőtechnológia használata a mindennapi életben gyakorlatilag megkerülhetetlen, alkalmazásával további előnyök jelentkeznek. Ezek:

- agility (gyorsaság): széles körben hozzáférhető technológiák, így gyorsabban, könnyebben lehet bevezetni innovatív technológiákat. Gyorsan lehet szolgáltatásokat terjeszteni.
- elasticity (rugalmasság): terhelés függvényében gyorsan lehet változtatni az erőforrásokon. Az erőforrások szabadon skálázhatók felfelé, lefelé, az épp igényelt terhelésnek megfelelően.
- cost saving (költségcsökkentés): a felhőtechnológiák használata lehetővé teszi, hogy kiszámíthatóak és előretervezhetőek legyenek a költségek.
- deploy globally in minutes (globálisan gyors terjesztés): a felhőtechnológia segítségével új földrajzi helyek érhetők el nagyon gyorsan, akár percek alatt világszerte. Az alkalmazások folyamatosan fejleszthetők, a felhasználói gépeken mindig az aktuálisan legfrissebb verzió állhat rendelkezésre.

3.1 A felhőtechnológia kialakulása [11]

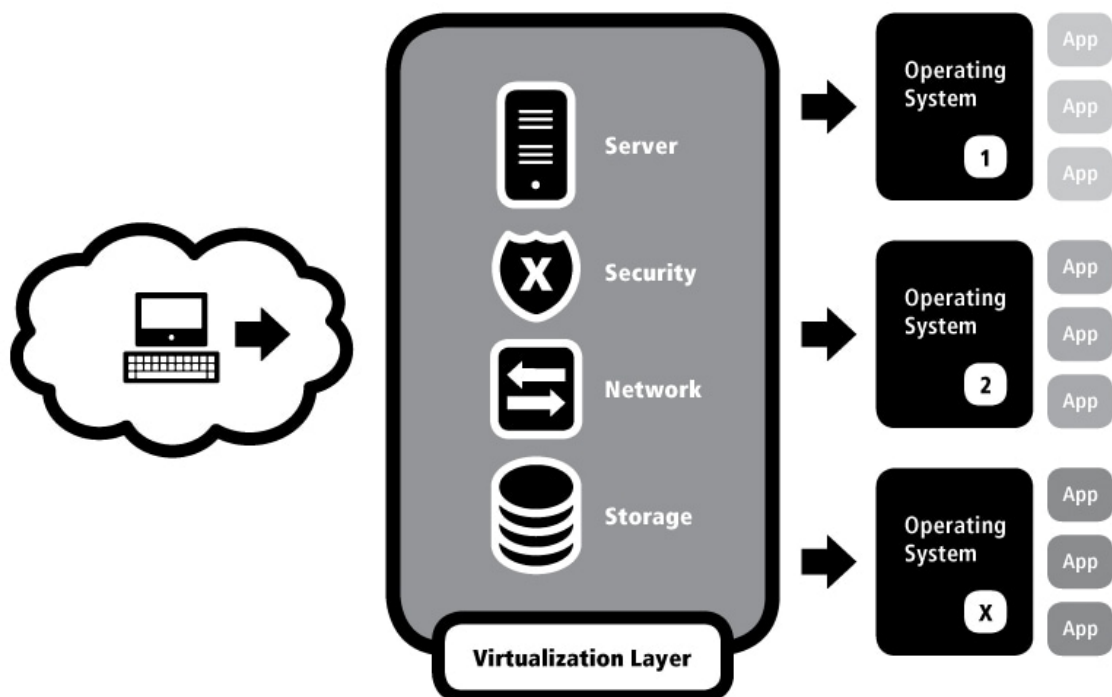
Nehéz határozott időpontot mondani a felhőalapú rendszerek kialakulására. Ennek részben az az oka, hogy a különböző publikációk nem értenek egyet a

felhőalapú szolgáltatások fogalmában. Az Egyesült Államok kormányának is gondot okozott a cloud computing fogalma, ezért a definíció elkészítésével megbízta a NIST-t (National Institutes of Standards and Technology). Az 1 hónappal később kiadott értelmezés előtt volt egy fontos bevezető mondat: „*a számítási felhő más és más dolgokat jelenthet és jelent is a különböző emberek számára*”. Egyes publikációk ismert helyen lévő szolgáltatást biztosító rendszer esetében is felhőalapú szolgáltatásról beszélnek. Más publikációk az outsourcing (erőforrások kiszervezése) megjelenéséhez kötik a felhőalapú szolgáltatások megjelenését. Ez részben elfogadható, hiszen egy helyesen értelmezett felhőalapú szolgáltatás valóban lehet kiszervezett, ugyanakkor például egy cég belső hálózatán kialakított, például felhőalapú tárolás nem kiszervezett szolgáltatás. Tény, hogy a felhőalapú szolgáltatások megjelenéséhez (annak teljesítményével ebben a szakaszban még nem foglalkozva) sok előzetes feltétel teljesülése szükséges (például nagy sebességű számítógépes hálózat, számítógépes hálózatot támogató operációs rendszer, és az operációs rendszerben meglévő, de attól részben független szolgáltatások). A felhőalapú szolgáltatások időbeli megjelenését érdemes tehát annak folyamatában áttekinteni.

A felhőszolgáltatásoknak jelenleg nincs konkrét definíciója. A legtöbb publikáció inkább körül járja az idetartozó fogalmakat, de nem ad meg egzakt megfogalmazást. Szinte teljes egyetértés van a működéshez szükséges számítógépes hálózat esetében. Az áttekintett publikációk mindegyike az Internetre alapoz, amelynek kialakulásában meghatározó szerepe volt a DoD által kezdeményezett és finanszírozott ARPANET projektnek, illetve az ugyanilyen nevet viselő hálózatnak, az ARPANET-nek, amelynek egyik alap protokollja a TCP/IP. A következő mérföldkő több publikáció szerint is (köztük természetesen az IBM hivatalos oldala) az 1970-ben a System/370 mainframe számítógépre tervezett VM 370 (Virtual Machines) operációs rendszere. Ilyen számítógépet üzemelttek be a SZTAKI-nál 1979-ben, amelyen távolról, hálózaton keresztül, „buta” terminál segítségével közel egy évig dolgoztam. A VM 370 operációs rendszer (bevezetve és implementálva a virtualizációs réteg fogalmát) lehetővé

tette, hogy 1 fizikai gépen több virtuális gép működjön a fizikai memóriánál több virtuális memóriát biztosítva.

A versenytársak is kifejlesztették saját virtuális gépes rendszerüket, és az egyes virtuális gépeken akár különböző operációs rendszereket futtatva (11. ábra) biztosították a különböző alkalmazások szétválasztásának lehetőségét. Tág teret kapott a virtualizáció.



11. ábra A virtualizációs réteg szolgáltatásai

[Forrás:[22]]

Ugyanerre az időre tehető a cluster-ek megjelenése. A mainframe számítógépekkel ellentétben itt már nem 1 nagy teljesítményű számítógép végezte a számításokat (igen jelentős beszerzési ár mellett), hanem kisebb teljesítményű számítógépek együttműködésének kialakításával létrehozták a **cluster computing**-ot. A több kisebb számítógép képes volt ugyanazt a számítási teljesítményt biztosítani jelentősen alacsonyabb bekerülési és működtetési költségek mellett. A cluster alapját adó egyéb feltételek (hálózat, buszrendszer) azonban a mainframe számítógépeknél korlátosabb teljesítményt biztosítottak.

Egyetértés van a következő mérföldkő esetében is, amikor is az 1990-es évek elején egyes távközlési cégek dedikált pont-pont kapcsolatot kínáltak ügyfeleiknek. Ezzel megnyíltak a VPN-re (Virtual Private Network) épülő lehetőségek. Szintén a 90-es években mutatkozott be a **hypervisor**, amelyet a fizikai gépekre telepítve, és konfigurálva lehetőség nyílt a fizikai eszközöktől elszakadva egy olyan számítógéprendszernek a megjelenítésére, amely egy egységként tüntette fel ezt az egész rendszert. Újabb számítógépek hozzáadása, illetve a rendszer részét képező számítógépek kivétele a hypervisor segítségével jelentősen leegyszerűsödött.

Ugyancsak a 90-es években jelent meg a cloud computing közvetlen elődjének számító **grid computing**, amelyet földrajzilag eltérő helyen lévő cluster-ek összekapcsolásával hoztak létre. Ez már igényelte a nagy távolságú számítógépes hálózatok jelentős sebességnövekedését, és robusztusságot.

Az 1997-es évben lehetett először találkozni a „**Cloud Computing**” fogalmával, amit Prof. Ramnath K. Chellapa vezetett be egy Dallas-ban megrendezett konferencián. Iparági kontextusban a cloud computing fogalmának bevezetését több publikáció, weboldal is 2006-ra teszi, amikor a Google vezérigazgatója kijelentette, hogy az *„adatszolgáltatásoknak nem a szervereken, hanem valahol a felhőben kell lenniük.”*

2006-ra gyakorlatilag megteremtődtek a feltételei a cloud computing-nak. A következő dátum (2006. márciusa) már az első olyan szolgáltatás megjelenése, amely ezekre építve megbízható eléréssel biztosította azt a felhasználók számára. Ekkor jelentette be az Amazon a Simple Storage Service (Elastic Compute Cloud) felhőalapú szolgáltatását.

3.2 Szolgáltatási modellek, felhőarchitektúrák

Szolgáltatási modellről 3, akár egymásra épülő rétegben lehet beszélni. A felhasználó ezek közül bármelyiket igényelheti, akár többet is. Közös jellemzője mindegyiknek, hogy a felhasználó az igényelt szolgáltatást on demand, azaz használat alapján fizeti. Így a költségek előre tervezhetők.

3.2.1 Infrastructura as a Service (IaaS)

Ez a legalacsonyabb rétegben lévő szolgáltatás. Leegyszerűsítve úgy lehet rá gondolni, mint a nyers hardware-re. A felhasználó megadja azokat a paramétereket, amelyekre szüksége van, lényegében úgy, mint amikor egy megvásárolni kívánt számítógép konfigurálása történik. Előnye ennek, hogy a felhasználónak nem kell befektetnie a hardware-be, nem kell törődnie a fizikai üzemeltetéssel, szükség esetén csökkentheti/növelheti az igényelt teljesítményt. Kööttségek lehetnek, és vannak is. Nem egy szolgáltató IaaS-t csak x86 platformra tud biztosítani. Az IaaS-nek része lehet a szükséges számítógépes hálózat is.

3.2.2 Platform as a Service (PaaS)

Szoftverplatform biztosítása szolgáltatásként. A platform több szinten közelíthető meg. A „nyers vas” működtetéséhez szükség van telepített, konfigurált operációs rendszerre (például Microsoft Windows Server, különböző Linux disztribúciók). Ezzel a legegyszerűbb platform már rendelkezésre áll. ugyanakkor a szolgáltatók jelentős része itt nem áll meg. Platformként kínál telepített (szükség esetén előre konfigurált) adatbáziskezelőt (például MqSQL, Microsft SQL Server, PostgreSQL), webservert, vagy a webservert egyes komponenseit (például Apache, Tomcat, Joomla, php), illetve levelező szervert (például Microsoft Exchange Server). A szolgáltatás előnyei többek között, hogy nem a felhasználó szerzi be a működtető (rendszer) szoftvereket, és megadhat olyan elvárásokat is, mint a rendelkezésre állás megbízhatósága. Nem kell sem rendszergazdáról gondoskodnia, sem a működtető szoftver rendszeres frissítéséről.

3.2.3 Software as a Service (SaaS)

A szolgáltató egy számítógépes hálózaton keresztül (lényegében Internet eléréssel rendelkező ügyfeleknek) szoftverhasználatot biztosít szoftverbérlési

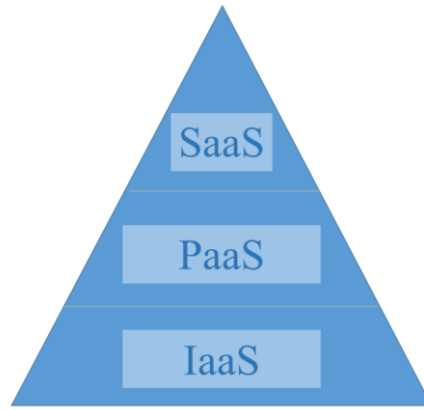
konstrukcióval. A szoftverek igen változatosak lehetnek. Kevés kivétellel (főként hardverhez szorosan kötődő alkalmazások) bármilyen területen érhetők el ilyen szoftverek:

- Irodai szoftverek (pl.: Microsoft Office 365, Google Docs, Sheets, Slides, ...);
- online filekonverterek (pl.: pdf, docx, képformátumok);
- lakásberendező szoftverek (pl.: RoomStyler, HomeByMe);
- előre konfigurált Webáruházak (pl.: SquareSpace, Shopify);
- számlázóprogramok (pl.: Számlázz, Billingo, BUPA);
- könyvelő programok (pl.: SmartBooks), stb.

Az előbb példaként említett szoftvereknek sok esetben létezik hagyományos, a felhasználó számítógépére telepíthető változata is (például a Microsoft Office 2021). Az SaaS-nek köszönhetően azonban a felhasználók (bérlők) mindig az aktuálisan legfrissebb szoftver verziót fogják használni, és a költségek itt is előre tervezhetők.

3.2.4 Felhő architektúrák

Annak ellenére, hogy a felhő alapú szolgáltatások már több éve elérhetők, még mindig nincs közös álláspont akkor, amikor a felhő architektúrák kerülnek szóba. Létezik olyan megközelítés, amely szerint a szolgáltatási modellek egyfajta architektúráként is értelmezhetők. Ezzel részben egyetértek, hiszen valóban jól látszik a modellek egymásra épülésének lehetősége (12. ábra). Ha a felhasználó egy PaaS keretében Windows Server-t bérel, az nem képes működni megfelelő hardware (IaaS) nélkül.



12. ábra Felhőszolgáltatási modellek egymásra épülése

[Forrás:[Saját szerkesztés]]

Lényegében ehhez hasonló a megközelítése az InterVison szoftverfejlesztő cégnek, akik elméletüket 2010-ben publikálták először, majd 2021-ben frissítették. [23]. A számítási felhőt üzemeltetői szempontból vizsgálva (bár kiemelik, hogy a felhőről rétegek szintjén nem korszerű beszélni) a következő 5 réteget nevezik meg:

1. Cloud Application
2. Cloud Software
3. Cloud Infrastructure
4. Kernel
5. Hardware

Az első három réteg sorrendben fentről lefelé haladva, jól megfeleltethető a korábban bemutatott SaaS, PaaS, IaaS szolgáltatásoknak. Az IaaS-t a felhasználó virtuálisan érzékeli ugyan (igényelt hardware szolgáltatásként), de valójában azt egy vagy több fizikai számítógépből a felhőszolgáltató állítja össze, így belátható, hogy kell egy legalsó, egy kézzel fogható fizikai réteg, a hardware. Ugyanakkor mindegyik számítógép hardware-e esetében van egy azt közvetlenül működtető software, aminek segítségével ez a fizikai számítógép konfigurálható, egyes alaplapon lévő eszközök engedélyezhetők, vagy tilthatók, stb. Ennek a fizikai réteg felett kell elhelyezkednie.

Véleményem szerint ide tartozik megközelítéstől függően a hypervisor, amely ezt a hardware-t teszi elérhetővé a felhő számára. Ez utóbbi két réteggel a felhő felhasználója (aki egy vagy több szolgáltatás bérlője) elviekben sosem találkozhat.

3.3 Felhőszolgáltatások

Több elterjedt felhőszolgáltatás létezik, és nem egy esetben ezek összefonódnak. A COVID idejében nagyon gyorsan kellett átállni az oktatási intézményeknek az online oktatási formára. Ennek lett az a következménye, hogy ugyanazon iskolán belül is a különböző oktatók más és más online oktatási szoftvert használtak. Az idő múlásával vagy az iskola vezetősége, vagy az érintett kormányzati szervek előírtak, javasoltak egy platformot. Így lett viszonylag sok felhasználója a Google Classroom-nak, vagy akár a Microsoft Teams-nek.

A Google Classroom és a Microsoft Teams esetében lehetőség van file-oknak tárhelyre való feltöltésére, hogy azt a diákok később elérhessék. Ez a tárhely valójában egy felhő alapú tárhely, és így a konferencia szoftver keretén belül egy felhő alapú tárolási szolgáltatás is elérhetővé vált.

Fontos abból az irányból is megközelíteni a felhőszolgáltatásokat, hogy azokat kik érhetik el. Ennek megfelelően a következő változatokat különböztethetjük meg:

- personal cloud (személyes felhő)
- private cloud (magán felhő)
- community cloud (közösségi felhő)
- public cloud (nyilvános felhő)
- hybrid cloud (hibrid felhő)

Personal cloud: 1 személy bérel egy felhőszolgáltatást, és megfelelő azonosítás után csak a bérlő személy fér hozzá. Tipikus eset amikor a Dropbox, a

OneDrive vagy például a Google Drive előfizetéssel rendelkező személy csak saját maga által elérhetően tárolja ott a file-jait (családi fényképek, scan-elt iratok, stb.). Egyes file-okat megoszthat mások által elérhetően csak olvashatóan, vagy akár szerkeszthetően is.

Private cloud: egy szervezet fér hozzá a szolgáltatáshoz, de a szervezetnek akár minden alkalmazottja. Jogosultságok adhatók és vonhatók meg. Sok esetben a szervezet telephelyén belül van a felhőszolgáltatást biztosító hardware.

Community cloud: több szervezet több alkalmazottja használhatja a felhőszolgáltatásokat. Jogosultságok adhatók és vonhatók meg. A szolgáltatás háttérét biztosító hardware lehet a felhőszolgáltatónál, illetve egyik vagy másik szervezetnél.

Public cloud: több előfizető által elérhető felhőszolgáltatás. A szolgáltatást biztosító hardware minden esetben a felhőszolgáltatónál van. Ez a felelős a megfelelő engedélyek kiadásáért.

Hybrid cloud: átmenetinek tekintett felhőszolgáltatási forma. A szervezet meglévő eszközeit is bevonva a szolgáltatásba egy nyilvános felhővel kiegészítve biztosítja a szolgáltatásokat az alkalmazottak, esetleg ügyelek részére.

Összefoglalva az eddig áttekintett technológiákat és modelleket, amelyeket saját véleményemmel is kiegészítettem a 3.1- 3.2 alfejezetekben, kijelentem, hogy az értekezés második tézisében az elkészítésre kerülő architektúrát a felhőtechnológiára kell alapozni.

A felhőarchitektúrára épülő moduláris tervezéssel implementációs szinten is foglalkozó publikációt nem találtam a szakirodalom kutatása során. Ehhez legközelebb egy olyan esettanulmány [13] publikációja áll, mely hipotézisem szerint igazolja azt az állításomat, hogy az architektúrának felhőtechnológiára kell épülnie. A következő fejezetben ezt a publikációt fogom két oldalban áttekinteni.

3.4 Felhőalapú gyártás

A Wei Wei, Feng Zhou, Feng-Fei Liang szerzőhármas a „Product platform architecture for cloud manufacturing” c. publikációjában [13] három pekingi telephelyű gyártócégre alapozva azt vizsgálta meg, hogy a felhőalapú gyártásnak van-e létjogosultsága. A bevezetőben tényként kezelik, hogy a *„felhőalapú gyártás egy új gyártási paradigmaként és integrált technológiaként jelenik meg. A felhő alapú gyártás egy új gyártási modell, amely épít az Internetre, az IoT-re (Internet of Things), és további különböző hálózatokra”*. Ehhez a már széles körben használt felhőalapú szolgáltatásokat (IaaS, PaaS, illetve SaaS) alapul véve, *új gyártási modell kifejlesztésével egy új felhőalapú szolgáltatást javasolnak, az MaaS-t (Manufacturing as a Service). A felhőalapú gyártás kulcsa, hogy létre kell hozni egy felhőalapú gyártás nyilvánosan elérhető termék platformját, és így meg lehet valósítani a gyártási erőforrások és lehetőségek egységes kezelését. Ehhez a felhőalapú gyártási szolgáltatáshoz a felhasználók igényeik szerint a felhőn keresztül bárhol, és bármikor hozzá tudnak férni.*

Megállapítja, hogy a jelenlegi felhőszolgáltatások használatához szükséges infrastruktúra már rég rendelkezésre áll, és a felhasználók is rendelkeznek olyan alkalmazásokkal, amelyekkel ezeket a felhőalapú szolgáltatásokat igénybe tudják venni. Ezt tovább gondolva kijelenti, hogy *kiemelten fontosak azok a technológiák, amelyekre ezek az alkalmazások építenek, mert így biztosítják a felhőalapú szolgáltatások használatát.*

Példaként hozza fel, hogy a Boeing már 2012-ben megvalósított egy olyan tervező keretrendszert, amely Interneten keresztül működött együtt a több mint 40 országban levő, és a gyártásban résztvevő gyártócégekkel, aminek eredményeképpen az így elkészült Boeing 787-es utasszállító repülőgép esetében a fejlesztési idő is (30%-kal), és a fejlesztési költségek is (50%-kal) jelentősen csökkentek. Bár ez a rendszer még nem felhőtechnológiai alapon működött, de bizonyította, hogy a számítógépes hálózatot használva az elosztott rendszerek nagy hatékonysággal rendelkeznek.

Összefoglalom a publikációban bemutatott felhőtechnológiára épülő gyártási modelljük jellemzőit (ellentétben a korábban elterjedten használt rugalmas gyártási, valamint a hálózatra épülő gyártási modellel szemben):

- a gazdaságos erőforrás megosztás;
- a nyílt rendszerre épülés;
- a software, hardware, logisztika, és ismeretek felhőalapú elérése;
- a Petabyte mennyiségű adatáramlás;
- a valós használaton alapuló dinamikus konfigurálás;
- a lehetőség, hogy szinte bármilyen iparágból kapcsolódhatnak be cégek;
- a felhőszolgáltatások, az IoT használhatósága.

Fenti jellemzők véleményem szerint teljes mértékben megegyeznek az általam is megfogalmazott felhőtechnológia által biztosított előnyös tulajdonságokkal. Indokoltnak tartom a felhőtechnológia alkalmazását erre a feladatra.

Modelljüknek megfelelően elkészítettek egy implementációt, amely 3 pekingi gyártóüzem esetében kerültek bevezetésre. és különböző összetettségű gyártástervezési feladatokra alkalmazták. Két jellemzőt vizsgáltak:

- az első: a feladat hozzárendelési időszüksége;
- a második: a feladat elvégzésének időszüksége.

Méréseik szerint a feladat hozzárendelési ideje egyszerű feladatok esetében gyorsabb volt a hagyományos tervezési modell alapján működő rendszerekben, de amint összetettebb feladatokat kellett elvégezni, a felhőalapú modell egyre hatékonyabbnak bizonyult. A legösszetettebb feladatok esetében közel kétszer olyan gyors volt, mint a hagyományos. Jellegében hasonló eredményt kaptak a feladat elvégzésének idejét vizsgálva is, azzal a jelentős különbséggel, hogy csak a közepesen vagy annál összetettebb feladatok esetében volt a felhőalapú modell eredményesebb.

Kijelentem tehát, hogy a felhőalapú technológiát alkalmazva az esetek jelentős részében hatékonyabb volt a felhőtechnológián alapuló modell. Indokolt a felhőtechnológia alkalmazása.

4. A kidolgozandó rendszer

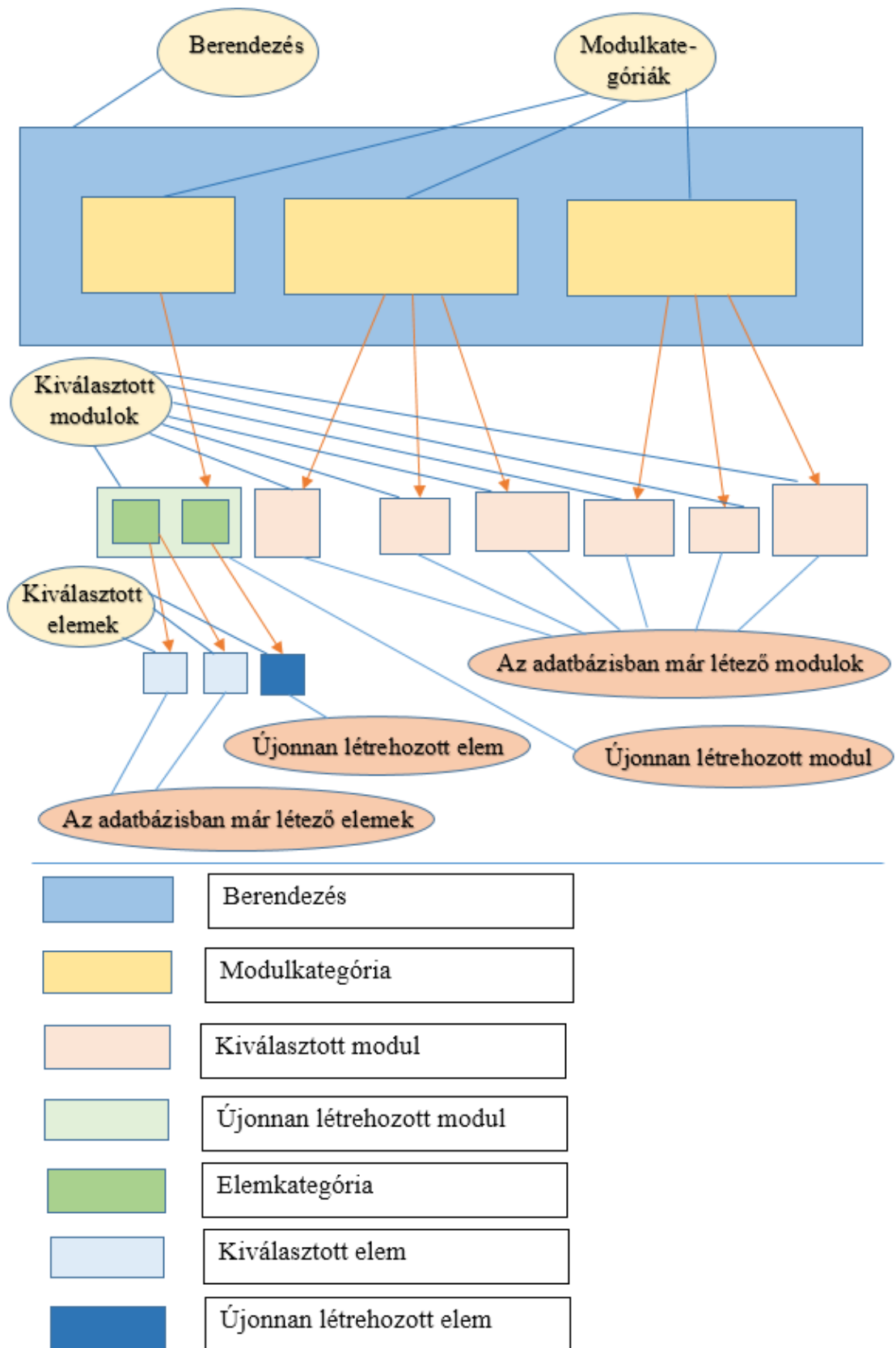
Összefoglalva a korábbi fejezetekben leírt állításaimat:

- indokolt egy építőelemes berendezés tervezésére alkalmas modell kidolgozása, mert hatékonyabb modell készíthető ennek alapján;
- indokolt egy olyan felhőtechnológiára épülő architektúra kidolgozása, amely alkalmas a fenti modell implementálására;
- indokolt egy olyan algoritmus kidolgozása, amely a modellen és az architektúrán alapulva hatékony felhőalapú, építőelemes tervezést végez.

4.1 Modell építőelemes berendezések tervezésére

Az értekezés egyik célkitűzése, hogy elkészítse a berendezés tervezésére alkalmas modellt. Ennek első lépéseként meghatároztam, hogy milyen modulokból állhat egy berendezés, és azt, hogy lehet ezeket a modulokat olyan általánosan meghatározni, hogy ne csak konkrét tervezési feladatok esetén legyen a modell használható. Ezt a gondolatmenetet követve eljutottam addig a legkisebb elemig, amelyet már nem fogok még kisebb összetevőiben vizsgálni. Ez nem jelenti azt, hogy a valós életben ezek az elemek ne lehetnének akár még kisebb alkotóelemeiben is vizsgálhatók.

Az értekezés első részében bevezetett fogalmakat használva készítettem először a berendezés átfogó, absztrakt képét, majd azt alapul véve, haladtam egyre alsóbb rétegek felé, és határoztam meg, hogy milyen módon lehet eltérő berendezések tervezésére alkalmassá tenni. Az elkészült modell a saját szerkesztésű 13-as ábrán látható.



13. ábra Az építőelem elvű (moduláris) tervezés modellje

[Forrás:[Saját szerkesztés]]

Az értekezésben célul tűztem ki, hogy a modellkészítés során olyan általános modellt készítsek, amely elviekben képes tetszőleges **berendezést** modellezni. Ehhez a top-down elvet használva határoztam meg a berendezésnek azokat a kisebb-nagyobb összetevőit, amelyekhez a számítógépes tervezés során konkrét értékeket fogok tudni rendelni.

Egy tetszőleges (fiktív) gyárat vizsgálva jellemzően egy gyártósor, vagy kisebb gyár esetében független-, vagy együttműködő gyártócellák vesznek részt egy berendezés összetevőinek gyártásában. Úgy határoztam meg a legmagasabb szinten lévő egységet, a **berendezést**, hogy a modell ne korlátozza ezt (a berendezést) se a részét képező nagyobb egységek számában, se azok típusában. Ezért a berendezést nem modulokra, hanem modulhelyekre fogom vizsgálni, ehhez pedig bevezetek egy új fogalmat, **modulkategóriákat**. Az ugyanolyan feladatot szolgáló egyes berendezések oly mértékben térhetnek el egymástól, hogy a modulkategóriáknak teljesen általánosnak kell lenniük, de a tervezés első lépéseként a *berendezés típusának függvényében* ezek a modulkategóriák pontosításra kerülnek. Egy modulkategória esetén meg kell tudni fogalmazni a modulkategóriával szemben támasztott elvárásokat, amiket paraméterek segítségével kell a tervezőnek megadnia. A tervezés során a **modultárból** minden egyes modulkategóriához rendelni kell olyan **modulokat**, amelyek megfelelnek a támasztott elvárásoknak.

Amennyiben nem sikerül a modultárból megfelelő modult találni, úgy létre kell hozni egy új modult. Ehhez a modulkategóriát képező **elemhelyekre** kell megfelelő elemet keresni. Ennek első lépése az elemhelyeknek a meghatározása. Bevezetve egy új fogalmat, minden egyes modulkategória egy vagy több **elemkategóriából** fog állni. Az adatbázisban már meglévő elemek tárolási helye az **elemtár**. Az elemkategóriákhoz olyan **elemeket** kell rendelni, amelyek paramétereik alapján megfelelnek az elemkategóriával szemben támasztott elvárásoknak.

Amennyiben nem sikerül egyetlen megfelelő elemet sem találni, úgy bővíteni kell az elemtárat olyan új, az elvárásokat teljesítő elemmel, amely a megfelelő elemkategóriához hozzárendelhető.

Tézis I. Kidolgoztam egy olyan moduláris rendszerek tervezését támogató rendszermodellt, amely építőelv alapján biztosítja a modulok hatékony illeszkedési vizsgálatát, építve a már korábban elkészített rendszerekből kinyerhető információkra. Az absztrakt berendezések és absztrakt modulok rétegének elemei összerendeződnek a konkrét elemekkel, ahol a kapcsolatok az attribútumokon keresztül szabályokkal felügyeltek. A javasolt modell kiemelkedő rugalmasságot és újrahasznosíthatóságot biztosít.
[S1] [S3] [S17]

4.2 Felhőalapú építőelemes tervező program architektúrája

Az értekezésben kidolgozásra kerülő architektúra egyik fő célja az, hogy az architektúra minél több összetevője épüljön a korábban bemutatott felhőalapú szolgáltatásokra és használja az előző fejezetben tézisként bemutatott modellt.

A tervezés során a következő kérdéseket fogalmaztam meg:

1. Hány réteg legyen?
2. Milyen funkciókra van szükség, és azok milyen rétegekbe kerüljenek?
3. Az architektúra zárt, szigetszerű legyen vagy adjon lehetőséget más rendszerből történő adatok importálására?

Válaszok a kérdésekre:

1. *Célként határozom meg, hogy az architektúrának támogatnia kell a kliens/szerver elvet.* Ennek megfelelően kell lennie egy *kliens* és egy *szerver* rétegnek, vagyis két fő rétegnek. A szerver réteg minden

összetevője épülhet felhőszolgáltatásokra, így a szerver réteg három alréteget fog tartalmazni (SaaS, PaaS, IaaS).

2. Funkcionális elvárások, rétegek:

- a kliens oldalon lévő felhasználói PC-k legyenek képesek egymás közt adatokat cserélni;
- az architektúra támogassa a szoros felhőintegrálódást;
- a tervezői elvárásokat kiszolgáló modul legyen független a WebApplication-tól;
- legyen független biztonsági modul;
- támogasson többféle adatbázis kapcsolati middleware-t (ADO, JDBC);
- az egyes rétegekben lévő program összetevők implementációinak image-e a felhőben legyen tárolva;
- az adatbáziskezelő a felhőben, egy felhőrétegben legyen;
- az adatbázis maga is a felhőben legyen tárolva.

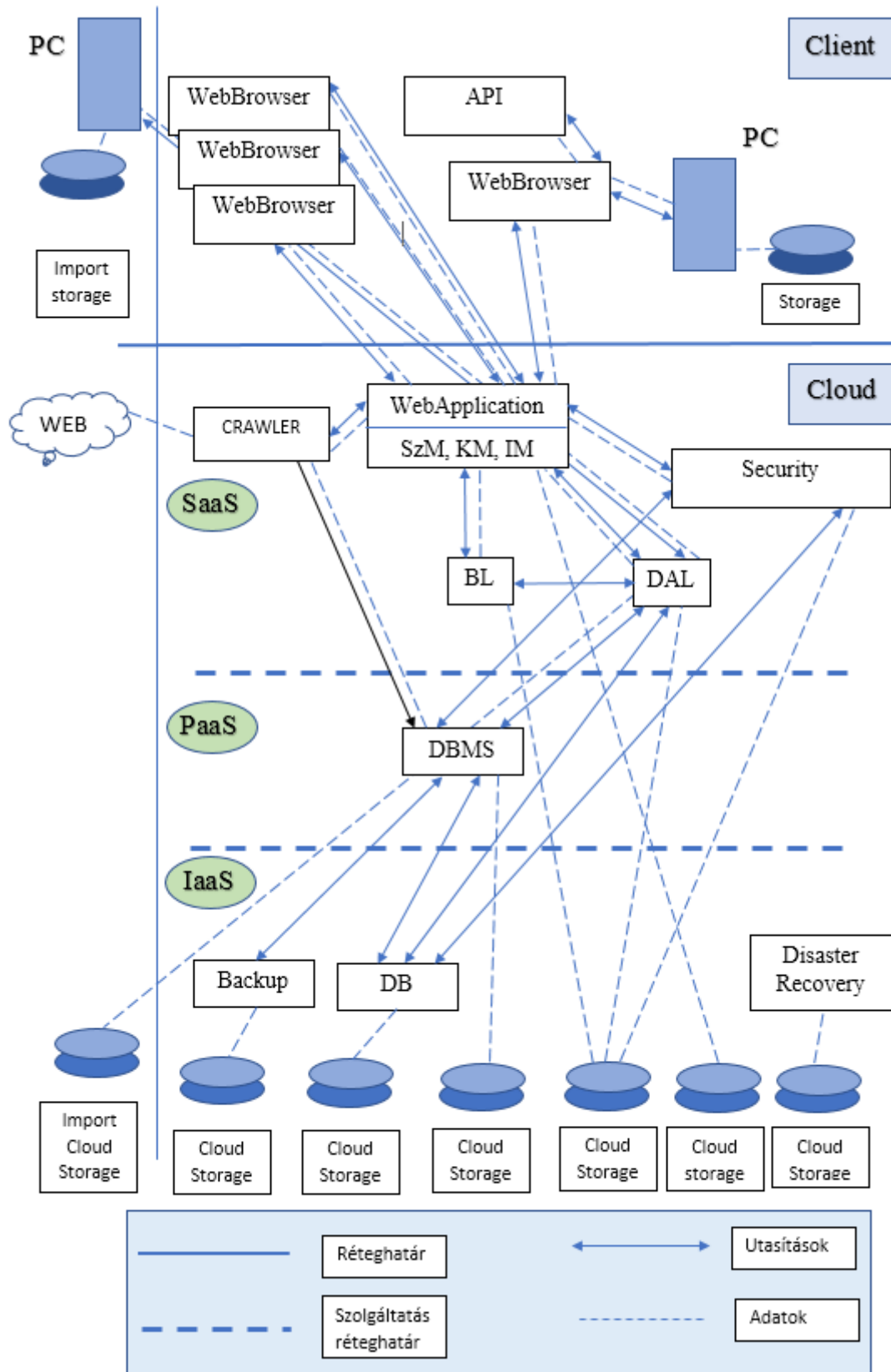
3. A korszerű informatikai rendszerek egyre inkább nyitottak, így a 3-as kérdésre a válasz, hogy az architektúra ne legyen zárt. Biztosítson lehetőséget más rendszerekből történő adatok importálására.

Célként fogalmazom meg, hogy egyidőben egymástól függetlenül, különböző földrajzi helyeken lévő tervezők képesek legyenek az alkalmazást használni. Ehhez a kliens oldalon csak egy UI lesz, minden más a felhőben.

Célul tűzöm ki, hogy a kialakításra kerülő architektúrára építve más felhőalapú alkalmazások is készülhessenek, valamint hogy az architektúra nyílt legyen.

Az elkészült architektúrát bemutató 14. ábrán a szerver réteg Cloud néven lesz beazonosítható. Az ehhez tartozó felhőszolgáltatási rétegeket korábban bemutattam.

A megfogalmazott kérdéseket figyelembevéve a következő architektúrát alakítottam ki:



14. ábra A kidolgozott felhőarchitektúra

[Forrás:[Saját szerkesztés]]

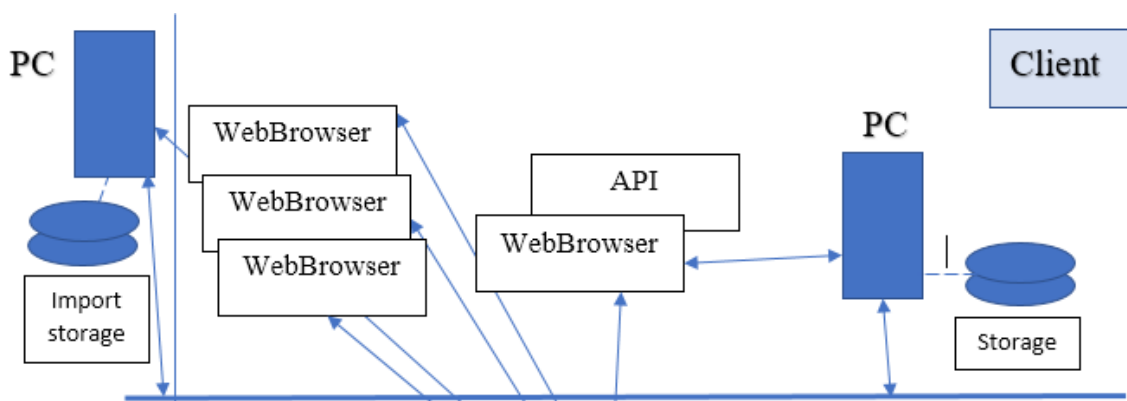
Felhőalapú alkalmazások sok változatban léteznek annak függvényében, hogy mennyire építenek azokra. A legegyszerűbb változatban az alkalmazás a felhasználó számítógépre van telepítve. Indítás után az alkalmazás minden adatot lokálisan tárolt file-okból olvas be, futás közben az átmeneti file-okat is ott hozza létre, majd a futás végén az átmeneti állományokat törölve csak a végeredményt tárolja egy felhőszolgáltatótól bérelt tárhelyen. Sok esetben a szolgáltató készít, vagy készíttet egy olyan OS kiegészítőt, amely a többi alkalmazás számára egy hagyományos meghajtóként emulálja a felhőszolgáltatónál bérelt tárhelyet, és azt vagy realtime, vagy adott időközönként szinkronizálja a felhőbe.

Költség- és erőforrás hatékonyság célból a lokális számítógépes rendszer által biztosított erőforrásoktól célszerű minél inkább függetlenedni. Minél jobban építenek a felhasználók, és az általuk futtatott alkalmazások a felhőszolgáltatásokra, annál kisebb mértékű a helyi kézzelfogható számítógépes rendszertől való függés. Ezzel egyidőben az alkalmazás fejlesztője is részesül a felhőszolgáltatások egy vagy több előnyéből. Ezt szem előtt tartva kezdtem megtervezni az architektúrát.

Az architektúrában felülről lefelé haladva, legfelül a felhasználó által indított alkalmazás kell legyen. Minden alkalmazás futásához szükség van erőforrásokra. Ezeket az alkalmazás készítője megoldhatja saját fejlesztéssel, és ekkor képes lesz direkt módon elérni a hardware-t (billentyűzetet, a grafikus kártyát), de ez nem kifizetődő, mivel a hardware gyártók a felhasználói igényeknek megfelelően nagyon sokféle hardware-t készítenek, és ezek szinte mindegyike egymástól különböző címzésekkel, megszakításokkal rendelkezik. Ezt a gyártók többnyire vagy a saját maguk által elkészített kis programok segítségével (drive) oldják meg (itt is külön figyelve arra, hogy milyen OS milyen verziójához kell azt kifejlesztteni), vagy egy másik, már korábban kifejlesztett eszközzel kompatibilis elérési módot tesznek lehetővé. Ez utóbbi általában kevésbé az adott eszközzel optimalizált, de jellemzően erősen költséghatékony hardware fejlesztést tesz lehetővé. Egyik tárgyalt megoldás sem járható út, kivéve a speciálisan egy hardware-re fejlesztett alkalmazásokat. Az operációs rendszerek megjelenésének

egyik fő oka épp az volt, hogy az OS által futtatott alkalmazás az erőforrásokat az OS segítségével fogja tudni elérni. *Összefoglalva az eddigieket: van arra mód, hogy a felhasználó által indított alkalmazás saját maga kezelje a hardware-t, de ezt el kell kerülni, az OS-re kell építeni.*

Következő eldöntendő kérdés, hogy az OS hol legyen. Hagyományosan az OS a felhasználó PC-jére kerül lokálisan telepítésre. A felhőszolgáltatások megjelenésével azonban jelentek meg olyan jól használható megoldások, amelyek esetében a valódi, többfelhasználós, multitask-os OS a felhőben fut. Ennek az OS-nek a szolgáltatásait úgy lehet igénybe venni, hogy többnyire egy speciális, lebutított működtető rendszer (egy célalkalmazás) fog először beboot-olni, amely csak az adott PC gyári hardware-ait képes kezelni. Ezután kell majd elindítani egy WebBrowser-t, amely lehetővé teszi a felhőalapú OS, és annak szolgáltatásainak elérését. Ennek előnye, hogy ez a lebutított működtető rendszer nagyon kisméretű, speciálisan az adott hardware-hez lett optimalizálva. Gyorsan boot-ol, kis erőforrás igényű, a PC saját BIOS-án keresztül konfigurálható. Ilyen megoldással lehetett találkozni több HP EliteBook-nál. Lényegében ez a PaaS-nek egy változata. A felhasználók jelentős része azonban rendelkezik korábbról olyan PC-vel, amelyre már telepítve van vagy egy OEM, vagy egy saját beszerzésű OS. *A kialakításra kerülő architektúra erre építem: a felhasználó saját PC-jére telepített OS lesz az alap, és az fogja futtatni a WebBrowser-t (15. ábra).*



15. ábra A kidolgozott felhőarchitektúra kliens rétege

[Forrás:[Saját szerkesztés]]

A továbbiakban a felhő által biztosított szolgáltatásokra építve tervezem meg az architektúra többi részét.

A felhasználó által indított WebBrowser egy WebApplication-nel kommunikál. Ennek felhőszolgáltatásokra kell épülnie, amit két szinten tud megtenni. Az első szinthez köthető, hogy a WebApplication tulajdonképpen önmaga is egy felhőszolgáltatás, amely a szolgáltató által biztosított SaaS rétegben helyezkedik el. A második szint akkor jelenik meg, amikor ez a WebApplication indításra kerül, mivel a futó szolgáltatásnak kell lennie egy image-ének, egy „lemezképének”, ahonnan indításakor betöltődik, relokálódik, majd futás közben hozzáfér a futásához szükséges erőforrásokhoz. Az image az IaaS rétegben lévő Cloud Storage-on kerül tárolásra.

A WebApplication 3 elkülönített program modult tartalmaz. Ezek:

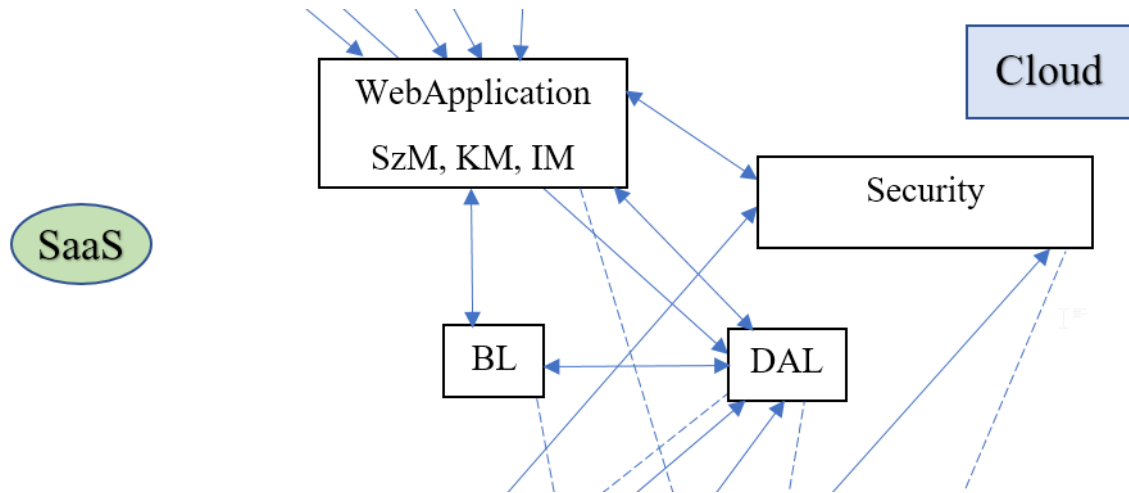
- a **Szakértői Modul** (SzM),
- a **Kereső Modul** (KM), valamint
- az **Illesztési Modul** (IM).

A **Szakértői Modul** a tervezési folyamat több lépésében kapja meg a vezérlést. A tervezőtől megkapott adatok alapján az adatbázisból, vagy az implementált tervezői számításokból állítja elő a futáshoz szükséges további adatokat, végez más modul által előállított eredményeken elemzéseket, illetve állítja össze a kiválasztott modulokból a kívánt berendezést. A **Kereső Modul** a megkapott paramétereknek eleget tevő modulokat, elemeket választja ki az adatbázisban már tárolt modulok elemek közül. Az **Illesztési Modul** a kiválasztott elemek, modulok esetén vizsgálja meg az összeépíthetőségét, és szükség esetén vizsgálja a soron következő kiválasztott elemet, modult, amíg az összeépíthetőség nem teljesül.

A felhasználó által indított WebBrowser egy **API-n** keresztül olyan szolgáltatásokat biztosíthat a tervező számára, amelyek segítségével tervrajzot vihet fel, vagy egyéb, a tervezéshez szükséges információt adhat meg.

Annak érdekében, hogy az egyes fejlesztési feladatok elválaszthatók legyenek, érdemes további, ugyan az SaaS-ben lévő, de nem a felhőhöz, hanem az alkalmazáshoz kötődő rétegeket kialakítani. Az egyik a BL (üzleti logikai réteg). Lényegében ebben a rétegben fogom elhelyezni azokat a fejlesztéseket, amelyek nem az alkalmazás futását biztosítják (nem „infrastruktúra” jellegű komponensei az alkalmazásnak), hanem amelyek a felhasználó által megfogalmazott igényeket szolgálják ki. Lényegében az adatrepresentációk, az algoritmus, az adatok közti relációk. Az üzleti logikát lehet tárolt eljárásokba is tenni. Ezeket egy további felhőszolgáltatási rétegben fogom elérhetővé tenni.

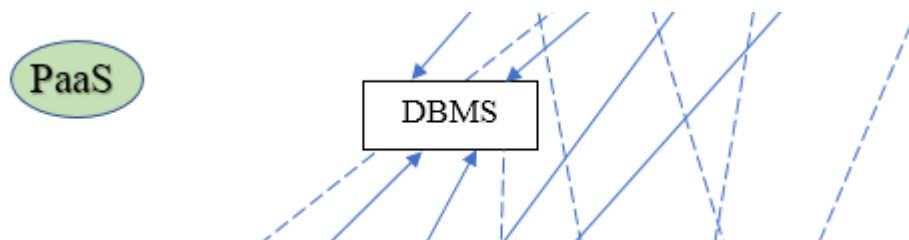
Ugyanebben az SaaS felhőszolgáltatási rétegben kell lennie még egy, az alkalmazáshoz kötődő rétegnek, a DAL-nak (az adatkapcsolati rétegnek). Egyrészt ez fogja lehetővé tenni, hogy az adatbáziskezelőnek „tetszőleges” módon lehessen lekérdezéseket küldeni, másrészt lehetővé teszi, hogy az adatbáziskezelőt le lehessen cserélni a WebApplication újra történő elkészítése nélkül. Nemcsak felhőalkalmazásoknál, hanem gyakorlatilag mindenhol, kiemelten szükség van egy lehetőleg független összetevőre, amelyben a biztonsággal kapcsolatos feladatok valósulnak meg (Security). Minden művelet esetében ki kell értékelni, hogy a szolgáltatáskérés jogosult felhasználótól indul-e, illetve maga a szolgáltatáskérés jogosult-e. Ennek a feladatnak az elvégzésére készíteni kell egy modult, amely futhat szolgáltatásként a felhőben, így ennek is ebben az SaaS felhőszolgáltatási rétegben a helye. Ezeket összegezve a 16. ábrának megfelelően készítettem el az SaaS felhőszolgáltatási réteget:



16. ábra A kidolgozott felhőarchitektúra SaaS rétege

[Forrás:[Saját szerkesztés]]

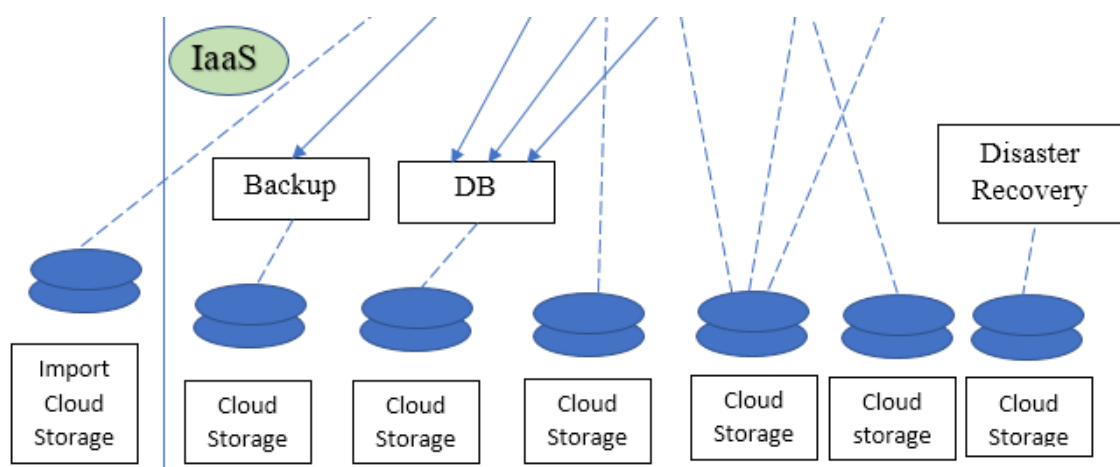
A következő felhőszolgáltatási réteg a PaaS, amelybe az építőelemes tervezéshez szükséges felhőalapú tervezés esetében egyetlen komponenst helyeztem el, ez az adatbáziskezelést végző DBMS (17. ábra). Futásához szükség van OS-re, amelyet a felhőszolgáltató tud biztosítani, de adott esetben akár magát az adatbáziskezelőt is. Így teljes mértékben lehet építeni a PaaS felhőszolgáltatási réteg által nyújtott lehetőségekre.



17. ábra A kidolgozott felhőarchitektúra PaaS rétege

[Forrás:[Saját szerkesztés]]

A működéshez szükséges alapok lényegében még hiányoznak. Ezeket az IaaS felhőszolgáltatási rétegbe lehet tervezni. Az eddigiek során a különböző rétegekbe tervezett architektúra elemeknek tárolódniuk kell valahol. Ezen architektúra elemek mindegyike használhat Cloud Storage-ot (felhőalapú tárolóhelyeket). Ugyanígy tárolni kell magát a fizikai adatbázist, és annak backup-jait, illetve a különböző vészhelyzetekre kidolgozott forgatókönyv által előírt lépésekhez tartozó fizikai mentéseket (Disaster Recovery) az alkalmazásokról, és azok beállításairól. A fentiek alapján az architektúrát a 18. ábrán láthatóan alakítottam ki:



18. ábra A kidolgozott felhőarchitektúra IaaS rétege

[Forrás:[Saját szerkesztés]]

Az alfejezet elején megfogalmazott kérdésekre válaszul készítettem el a felhőalapú architektúrát. Jelenleg azonban ez az architektúra zárt. Olyan tervezési feladat esetében, amikor nincs az adatbázisban a felmerült tervezési feladat elvégzéséhez megfelelő elem, akkor csak kézi adatbevitellel bővíthető az adatbázis. Ez lassú, hibalehetőséget rejt magában, és nem a mai elvárásoknak megfelelő. Ezért nyitni kell az architektúrán. Ki kell alakítani egy olyan import lehetőséget, amely több irányból is elérhetővé teszi a meglévő adatbázishoz új elemek hozzáadását. Jelenleg erre két módot látok. Az egyik egy PC-n keresztül, az ott lokálisan vagy a felhőben tárolt, de csak a PC-n keresztül elérhető elemek adatainak importját teszi lehetővé. A másik esetben ezek az adatok egy felhőszolgáltatón keresztül közvetlenül is elérhetők. Az importálás számos módon

valósulhat meg, vagy CVS-be, vagy XML-be történő előzetes exportálás után, de van mód akár a DBMS-sel kompatibilis adatbázis közvetlen elérésével. Az architektúra ezen része a 13. ábrán baloldalt a Cloud Storage, illetve Import Cloud Storage részeken követhető.

A fejezetben megtervezett felhőarchitektúra teljesíti mindazon elvárásokat, amelyek segítségével az építőelemes berendezések felhőalapú számítógépes tervezése megvalósítható.

Tézis II. Kidolgoztam egy új hatékony felhőalapú szoftver/hardver architektúra modellt az I. tézisben megadott elvű tervezési modell implementálására, így biztosítva a felhőalapú szolgáltatások előnyeit: platform függetlenség, hatékony szoftverfejlesztés/verziókövetés, helyfüggetlenség, hatékonyság. [S2] [S7-S9] [S18]

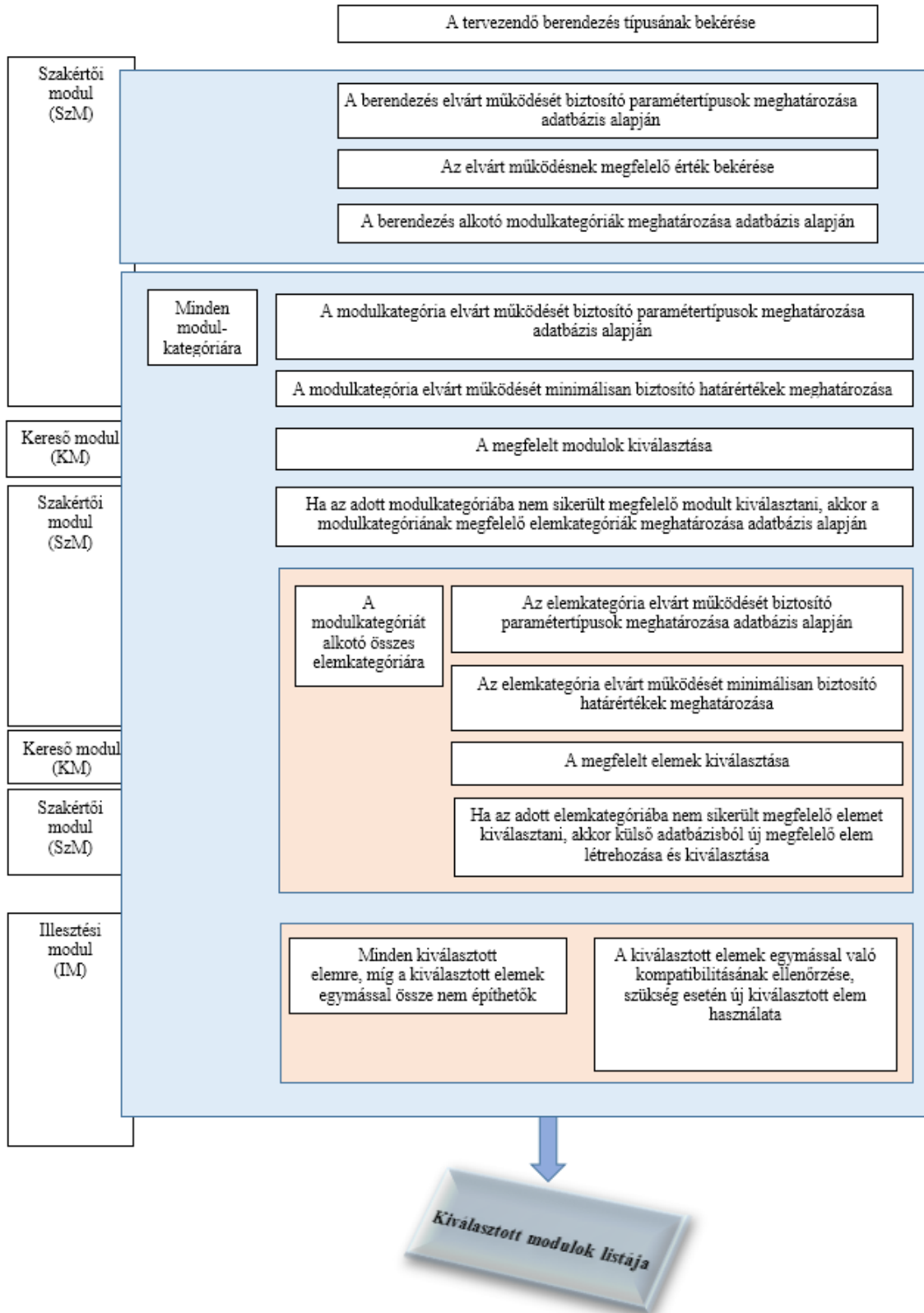
4.3 A moduláris tervezést végző algoritmus

Az értekezésben kidolgozásra kerülő algoritmus az előző fejezetben tézisként bemutatott architektúrára kell épüljön. Első lépésként megfogalmazom az elvárásokat az algoritmus felé:

- legyen képes elvben bármilyen berendezést tervezni;
- legyen képes a berendezés modul kategóriáit meghatározni;
- legyen képes a modul kategóriáknak megfelelő modulokat kiválasztani;

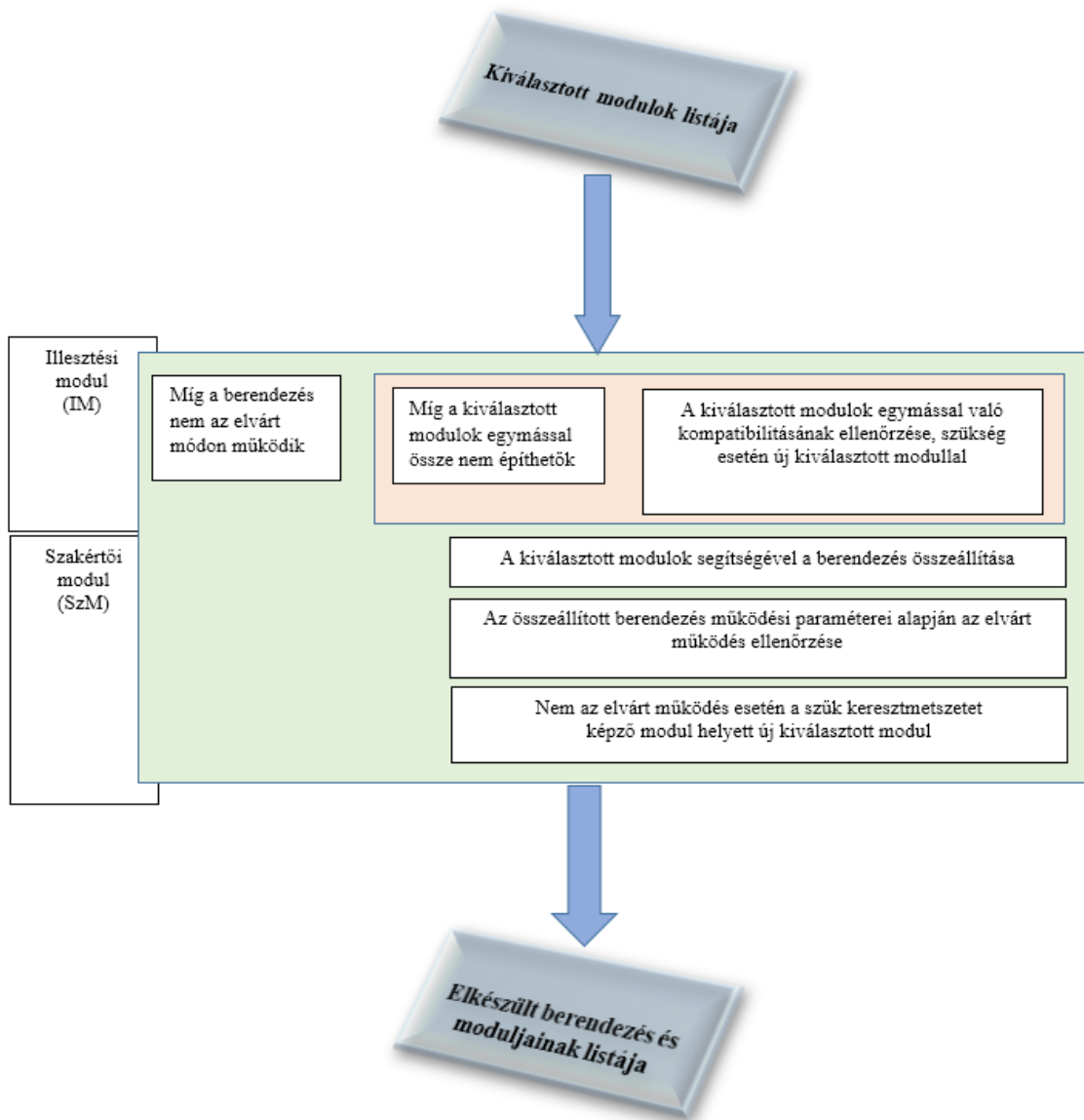
- legyen képes szükség szerint a modulkategória elemkategóriáit meghatározni;
- legyen képes az elemkategóriáknak megfelelő elemeket kiválasztani;
- legyen képes szükség szerint elemeket más adatforrásokból importálni;
- legyen képes a berendezés elvárt működését ellenőrizni, szükség esetén új kiválasztott modult használni.

A 19. és 20. ábrán látható algoritmust készítettem el, amely a fent megfogalmazott elvárásokat teljesíti:



19. ábra A modul kategóriákat kezelő algoritmus

[Forrás:[Saját szerkesztés]]



20. ábra A berendezés elvárt működését biztosító algoritmus

[Forrás:[Saját szerkesztés]]

4.4 Az algoritmus főbb lépései a következők:

- I. A tervező megadja a berendezést.
 1. Az adatbázis segítségével meghatározásra kerülnek a berendezés azon paramétertípusai, amelyek a berendezés elvárt működését biztosítják.
 2. A tervező megadja a berendezés elvárt működését meghatározó értékeket.
- II. Adatbázis segítségével meghatározásra kerülnek a berendezés modulkategóriái.
 1. Minden egyes modulkategória esetén az adatbázis segítségével meg kell határozni azokat a paramétertípusokat, amelyeknek konkrét értékei fogják meghatározni a modulkategóriába kiválasztásra kerülő modulokat.
 2. Meg kell határozni az előző lépésben kikeresett paramétertípusok azon határértékeit, amelyek az elvárt működést minimálisan biztosítani tudják.
 3. Tételiesen meg kell vizsgálni minden, az adatbázisban tárolt modult, hogy az megfelel-e a modulkategóriának, és paramétereinek megfelelnek-e az elvárt működést biztosító határértékeknek. Amennyiben igen, úgy a modul kiválasztásra kerül az adott modulkategóriához rendelve.
 4. Előfordulhat, hogy az adatbázis nem tartalmaz kiválasztható modult. Ebben az esetben elő kell állítani egy ilyen modult, majd az előállított modult ki kell választani.
 - i. Ennek első lépéseként adatbázis alapján meg kell határozni a modulkategória elemkategóriáit.
 - ii. Minden egyes elemkategória esetén az adatbázis segítségével meg kell határozni az elemkategória elvárt működését biztosító paramétertípusokat.

- iii. Meg kell határozni az előző lépésben kikeresett paramétertípusok azon határértékeit, amelyek az elvárt működést minimálisan biztosítani tudják.
 - iv. Tételesen meg kell vizsgálni minden, az adatbázisban tárolt elemet, hogy az megfelel-e az elemkategóriának, és paraméterei megfelelnek-e az elvárt működést biztosító határértékeknek. Amennyiben igen, úgy az elem kiválasztásra kerül az adott elemkategóriához rendelve.
 - v. Előfordulhat, hogy az adatbázis nem tartalmaz kiválasztható elemet. Ebben az esetben importálás vagy kézi adatfelvitellel az adatbázis bővítésre kerül egy az elvárásokat teljesítő elemmel, majd az elem kiválasztásra kerül.
5. Az elemkategóriákhoz kiválasztott elemeket (célfüggvény segítségével sorba rendezve, majd minden elemkategória esetén véve a lista soron következő elemét) meg kell vizsgálni, hogy az elemek valóban összeépíthetők-e. Ha nem, akkor a nem megfelelő elem esetében a listából venni kell a lista következő elemét, mindaddig, amíg a modult alkotó elemek össze nem építhetők.
6. Az eddigi lépések alkalmazásával előáll egy olyan lista, amely minden egyes modul kategóriához rendel egy vagy több kiválasztott modult.
- III. Az előző lépésben elkészült lista alapján egyesével venni kell az egyes modul kategóriákat, és célfüggvény segítségével a modul kategóriához rendelt modulokat sorba rendezve
1. minden egyes modul kategória esetén a modul kategóriához rendelt rendezett listából venni kell a soron következő modult.

2. Meg kell vizsgálni, hogy az ilyen módon összeállított berendezés teljesíti-e az elvárt működést.
3. Amennyiben nem, úgy a szűk keresztmetszetet adó modult le kell cserélni a modulkategóriához rendelt lista soron következő moduljával, mindaddig, amíg a berendezés nem az elvárt módon működik.

Az elkészített adatbázisban az egyik központi elem a modulok/építőelemek halmaza. Jelölje ezen listát a

$$\Gamma = \{M_i\}$$

szimbólum. Egy adott M_i modul/építőelem több egységből állhat, melyek mindegyike adott technológia paraméter listával rendelkezik:

$$M_i = \{A_{ij}\}, A_{ij} = \langle (p_k = v) \rangle$$

A továbbiakban jelölje

$$\Delta = \{A_k\}$$

az összes egység halmazát, ahol minden egység egy modulhoz kapcsolódik. A tartalmazó modult a

$$h : \Delta \rightarrow \Gamma$$

függvény jelöli. Az egység kategóriáját pedig a

$$t : \Delta \rightarrow T$$

függvény adja meg, ahol T a lehetséges kategóriák halmaza. Az egységekhez emellett egy költségfüggvény is rendelődik:

$$c : \Delta \rightarrow N^+$$

A felépítés során csak olyan egységek választhatók ki, amelyek egymással kompatibilisek.

A kompatibilitási reláció leírására az

$$R \subseteq \Delta \times \Delta$$

szimbólumot használom.

Egy új igény beérkezésekor meghatározásra kerül az illeszkedő modulok és az igényelt egységtípusok halmaza:

$$\Gamma_q \subseteq \Gamma$$

$$T_q \subseteq T$$

Ezek alapján a feladat azon

$$\Delta_q \subseteq \Delta$$

meghatározása, melyre

$$\forall t \ni T_q : \exists A \in \Delta_q : t(A) = t, h(A) \in \Gamma_q$$

$$\forall A_i, A_j \in \Delta_q : (A_i, A_j) \in R$$

$$\sum_{A \in \Delta_q} c(A) \Rightarrow \min$$

Ezen optimalizálási feladat megoldására az elemi brute-force megoldás egy

$$O\left(\prod_{t \in T_q} |\Gamma_{qt}|\right)$$

költségű algoritmust eredményez, hiszen minden lehetséges kombinációt meg kell vizsgálni.

Az alkalmazott optimalizálási módszer célja olyan egységek meghatározása, melyek egyrészt több más elemmel kompatibilisek és másrészt kis költségűek. Ez alapján célszerű egy jóság értéket rendelni minden egységhez az alábbi módon:

$$f(A) = \frac{|R_A|}{c(A)}$$

ahol az R_A szimbólum azon egységek halmazának a mértéke, melyek kompatibilisek az A egységgel.

Ezen jóság mérték alapján sorba rendezzük az A elemeket, és best first módszerrel keresünk megfelelő egységalmazt. A best first heurisztika egy többelemű megoldás esetén értelmezhető, ahol a módszer fő jellemzője, hogy egy adott rész megoldást az adott állapot szerinti legjobb lépéssel folytatjuk. A best-first optimalizáció széles körben alkalmazható különböző területeken, például mesterséges intelligencia, gépi tanulás, operációkutatás, játékelmélet, és döntéstámogatás.

Ez alapján a módszer lépései:

1. A kompatibilitási mátrix, a költségfüggvények meghatározása; a jóság értékek kiszámítása és az egységek e szerinti sorba rendezése; az igényelt egységtípusok jelzőinek alaphelyzetbe állítása.
2. A rész megoldások halmazának inicializálása egy üres halmazzal. Ezen halmaz egy heap struktúra, melyben az aktuális rész megoldások a költségük alapján kerülnek elhelyezésre.
3. A soron következő rész megoldás (legkisebb költség) kiemelése a heap tárolóból.
4. Az egységek tesztelése a jóság sorrend alapján; a tesztelésnél egy kompatibilitás ellenőrzésre és a típus vizsgálatára kerül sor. Ha találtunk illeszkedő egységet, akkor a rész megoldás aktualizálása, a költségének aktualizálása kerül sorra.
5. Az aktualizált (egységalmaz, költség) páros lehelyezése a rész megoldás halmazba. Az ide tartozó még szabad típusok listájának frissítése, a szabad típusok halmazának csökkentése a most bevont egység típusával.
6. Ha az aktuális rész megoldáshoz nem létezik illeszkedő egység, akkor ez az ág elhalt, nem kerül elem vissza a rész megoldások halmazába.

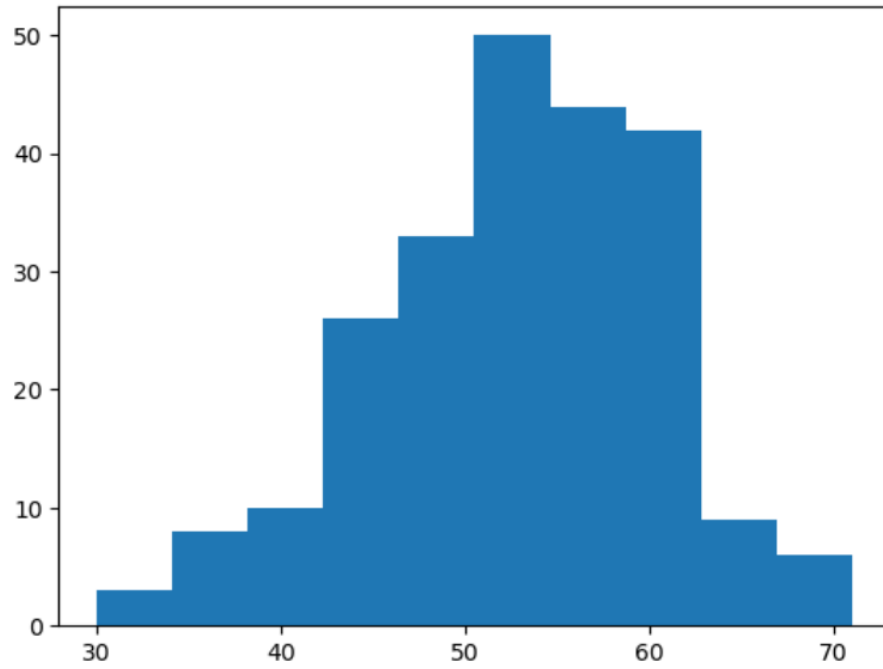
7. Ha az aktuális rész megoldásnál a szabad típusok halmaza üres lett, egy teljes megoldást kaptunk. Ez esetben sem kerül vissza új elem a rész megoldások halmazába.
8. A kezdeti paraméterek függvényében, több megoldást is kaphatunk, nem szükséges az első megoldásnál leállni. Mivel az egységek bevonása nemcsak a költségük alapján történik, emiatt nem garantált, hogy az első megoldás lesz a globális optimum. Emiatt célszerű egy nagyobb kísérletszámot megadni, azaz több illeszkedő összerendelést is meg kell vizsgálni.

Az algoritmus Python környezetben került implementálásra. A tesztek adathalmazait random, egyenletes eloszlású értékekkel állítottam elő. Mintaként egy

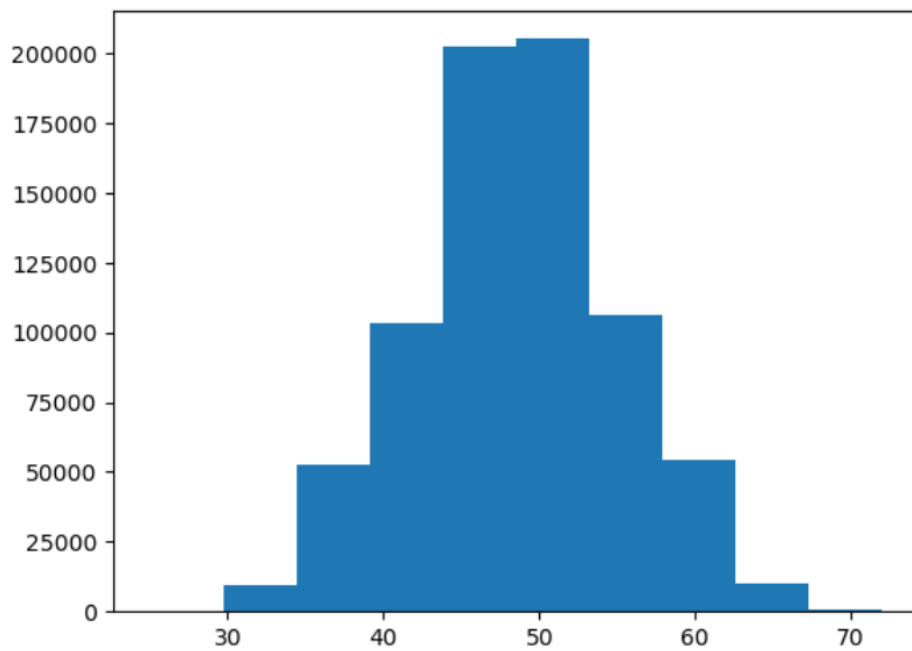
$$Tdb = 5, Adb = 42, Kp = .7$$

paraméterű eloszlást mutatok be, ahol a Kp paraméter a kompatibilitási reláció gyakoriságát adja meg, azaz a kapcsolatok ezen aránya esetében él a kompatibilitási reláció. A tesztben két módszer hatékonyságát vetem össze. Az egyik a teljes leszámolás módszere, a másik a javasolt optimalizálási algoritmus.

A tesztben a teljes leszámolást alkalmazva az egyes hozzárendelések költség szerinti hisztogramját is előállítom. A minta adathalmaz esetén a kapott hisztogramot az 21. ábra mutatja be. Az ábra x-tengelye mutatja a költségértéket, az y-tengely a gyakoriság értéket jelöli. A célfüggvény értékek 30-72 értékek között változnak.



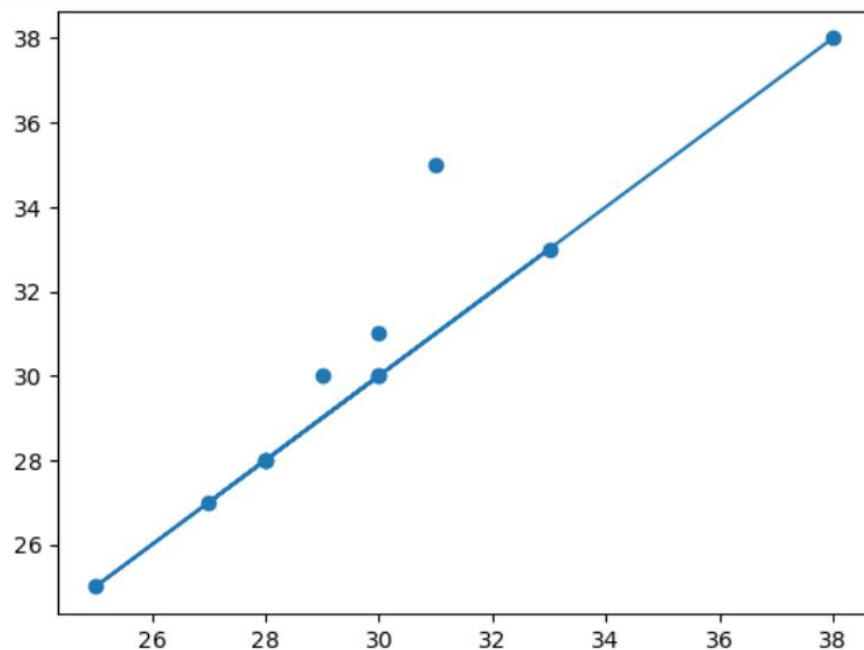
21. ábra Lehetséges összerendelések költség histogramja (Adb=56)



22. ábra Lehetséges összerendelések költség histogramja (Adb=210)

Az optimalizációs módszer által meghatározott lokális optimumok jósága a globális optimumokkal történő összevetésen keresztül mérhető. A globális optimumokat a teljes leszámolás módszere tudja szolgáltatni.

A tesztek alapján a javasolt optimalizáció az esetek 83%-ában globális optimumot ad. Ez elég jó közelítés, és az eredmény jóságát emellett az is bizonyítja, hogy a lokális optimumok átlagosan 98%-ban megközelítik a globális optimumokat.

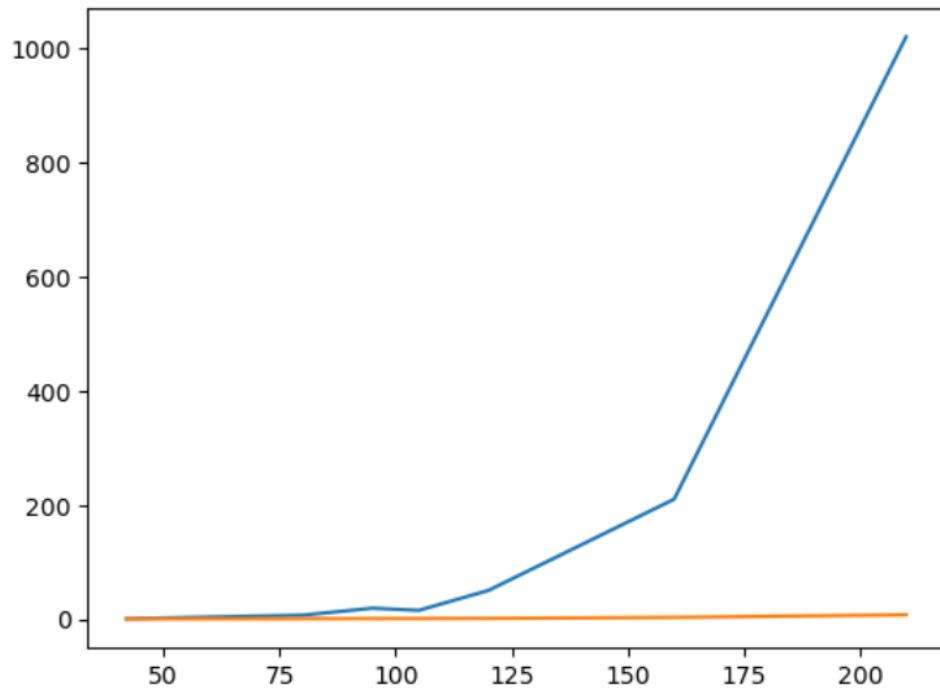


23. ábra Globális (x-koordináta) és lokális (y-koordináta) optimumok összevetése

A javasolt optimalizációs módszer fő előnye az időbeli hatékonyság. A javasolt módszer nagyságrendekkel gyorsabb, mint az alap brute force módszer, elsődlegesen a nagyobb méretű problémák esetén.

A 24-es ábrából látható, hogy $Adb = 200$ esetén a brute force időszükséglete 1000 sec, míg a javasolt optimalizációs módszernél csak 7 sec az időköltés.

A fenti mérések alapján a kísérleti eredmények jól mutatják, hogy a javasolt optimalizációs módszer jelentős javulást eredményez a brute-force alapmódszerhez viszonyítva, jól alkalmazható a nagyobb méretű problémák esetén is.



24. ábra Az optimalizációs és teljes leszámolás időkötségei

**Tézis III. A modulok kiválasztására kidolgoztam egy új algoritmust, amely a kijelölt célfüggvényhez meghatározza az optimális berendezés-
struktúrát. Az algoritmus több optimalizálási fázist tartalmaz az egyes rétegek elemeinek az összerendeléséhez. Az eljárás során meghatározásra kerülnek az optimális paraméter intervallumok, kompatibilitási feltételek. A megadott feltételek által meghatározott tartományon történik a kijelölt célfüggvény szerinti optimum meghatározása. [S17] [S19]**

4.5 Az algoritmus formális modellje

Jelölések:

B_i	Berendezés Az adatbázisban z db berendezés található. <u>Hivatkozás:</u> B_i , ahol $i = 1, \dots, z$
$MK_{i,j}$	Modulkategória MK_i a modulkategória mátrix-ának i -ik sora, vagyis az i -ik berendezés <u>Hivatkozás:</u> $MK_{i,j}$, ahol $i = 1, \dots, z, j = 1, \dots, y$, és y értéke egy tetszőleges i -ik sorban megegyezik $MK_{i,0}$ értékével.
y_i	az i-ik berendezés modulkategóriáinak darabszáma Minden berendezés esetén y más és más értékkel rendelkezhet, ezért értelmezhető az y_i , amely <u>Hivatkozás:</u> y_i , (ha van legalább 1 db modulkategória, akkor), $y_i = MK_{i,0}$
$PTB_{i,k}$	Berendezés paramétertípus i - az i -ik berendezés paramétertípusai <u>Hivatkozás:</u> $PTB_{i,k}$, ahol $i = 1, \dots, z$, $k = 1, \dots, x_i$, és $x_i = PTB_{i,0}$
x_i	Paramétertípus darabszám, <u>Hivatkozás</u> x_i , ahol i - az i -ik berendezés paramétertípusainak darabszáma

PTMK _{l,m}	<p>Modulkategória paramétertípusai</p> <p>PTMK_{l,1} - mindig az adott modulkategória elvárt működését biztosító paraméter típusa</p> <p><u>Hivatkozás:</u> PTKM_{l,m} ahol</p> <p>$l = MK_{i,1} , \dots , MK_{i,0}$</p> <p>$m = 1 , \dots , PTKM_{l,0}$</p>
PTMKH _n	<p>Modulkategória paramétertípusainak a berendezés elvárt működését biztosító paraméterértékből számolt határértéke</p> <p><u>Hivatkozás:</u> PTKMH_n ahol</p> <p>$n = 0 , \dots , MK_{i,0}$</p>
M _{o,p}	<p>A modulokat tartalmazó tábla</p> <p>tartalmazza a modulok adatait, köztük azt, hogy melyik modulkategóriához rendelhetők hozzá, illetve az adott modulnak mi a jellemző paraméterkategória értéke</p> <p><u>Hivatkozás:</u> M_{o,p} , ahol:</p> <p>$o = 1 , \dots , v$ (a modulok száma, amelyek az M táblában tárolva vannak)</p> <p>$p = 1 , \dots , a$ a modul nyilvántartott paraméterkategóriáinak száma</p>
KM _{q,r}	<p>Kiválasztott modulok táblája</p> <p><u>Hivatkozás:</u> KM_{q,r} ,ahol</p> <p>q - melyik modulkategóriához lett kiválasztva ($q = 1 , \dots , MK_{6,0}$)</p> <p>r – hányadik kiválasztott modul ($r = 1$, a megtalált esetek)</p> <p>KM_{q,r} – értéke a kiválasztott modul indexe az M modul táblából, ahol</p> <p>o a futó index</p>

<p>$EKMK_{s,t}$</p>	<p>A modulkategóriát alkotó elemkategóriák táblája</p> <p><u>Hivatkozás:</u></p> <p>$EKMK_{s,t}$, ahol</p> <p>$s = 1, \dots$, ahány modulkategória van</p> <p>$t = 1, \dots$, $EKMK_{s,0}$</p> <p>az s-edik modulkategória $EKMK_{s,0}$ elemkategóriából áll;</p> <p>az s-edik sor megadja az adott elemkategória további indexeit, amelyek a PTEK mátrixban használhatók</p>
<p>$PTEK_{i,j}$</p>	<p>Az elemkategória paramétertípusai</p> <p>i – az i-ik elemkategória paramétertípusai</p> <p>j - az i-ik elemkategória paramétertípusainak értékei és</p> <p>$PTEK_{i,0}$ – az i-ik elemkategória paraméter típusának darabszáma</p> <p>$PTEK_{i,1}$ –mindig az adott elemkategória elvárt működését biztosító paraméter típusa</p> <p>$PTEK_{i,2}$ – mindig az ár, így a célfüggvény segítségével könnyebb a hivatkozás</p> <p>$PTEK_{i,3,\dots}$ – az összes többi paramétertípus</p>

5. Mintarendszer gumihevederes szállítoszalagokra

A kidolgozott elmélet szállítoszalagok esetében való alkalmazhatóságának feltétele, hogy meg kell tudni adni a berendezés elvárt működését biztosító paramétereket, ezáltal a modulok kiválasztásához szükséges adatokat. Ehhez célul tűztem ki, hogy első lépésben meghatározzam, hogy egy szállítoszalag milyen modulkategóriákból épül fel. Alapvető elvárás az, hogy a berendezés kivitelezési és működtetési költségei minimálisak legyenek. Ez érvényes ebben az esetben is, a gumihevederes szállítoszalagok tervezésekor. Egy szállítoszalag beruházási költségei a következő elemekre bonthatók:

- 10-10%: egyéb tartozékok költségei
- 7-10%: villamos berendezések költségei
- 10-15%: acélszerkezet költségei
- 15-20%: görgők költségei
- 45-60%: a heveder költségei

A szállítoszalagtervezésénél kiemelten kell kezelni:

- a heveder jellemzők (szélesség, sebesség, szilárdság, stb);
- a görgők (görgőszám, görgőméret, minősítő paraméterek);
- a hajtási teljesítmény

meghatározását.

A szalagtervezés két változatával lehet találkozni:

- az egyedi, és
- az építőelemes.

Egyedi tervezés esetén a szállítoszalag minden egyes egységét (a heveder és a görgők kivételével) külön-külön megtervezik. Építőelemes tervezés esetén a rendelkezésre álló építőelemekből állítják össze a szalagkonstrukciót. Ebben az esetben belátható, hogy ugyanazt a szállítási feladatot többféle változattal (eltérő

gazdasági mutatók mellett) lehet megoldani. A szalag tervezése mindkét változat esetében sok számítást igénylő, összetett feladata. Ez indokolja a lehetőség szerinti minél automatizáltabb tervezési folyamatot. Az egyedi és az építőelemes számítógépes szalagtervezés moduláris felépítésű. A manuális tervezés jellemzően eltér ettől. Ez utóbbi esetben a döntések nem egy esetben szubjektív alapokon kerülnek meghozatalra, így nem garantálható, hogy a tervezés eredményeként optimális megoldás fog születni. Egyedi számítógépes tervezés esetén az egyes elemeknek számos változata állhat rendelkezésre egy konkrét tervezési modul használatakor. Az építőelemes (moduláris) tervezésnél elegendő az építőelemek terheléseinek (illetve az abból adódó jellemzők) meghatározása, és az így előállt elvárásoknak megfelelően a korábban ismertetett módon kikereshetők azon modulok, amelyek alkalmasak a berendezés elvárt módon történő működtetésére. Feladatomból volt olyan építőelemes berendezés tervezésének kidolgozása, amely alkalmas a szükséges berendezés nem a manuális, esetlegesen a szubjektív szempontokat figyelembe vevő moduljainak meghatározására.

Az elkészült módszer alkalmas moduláris, interaktív módon történő szalag megtervezésére, és ennek megfelelően:

- korszerű méretezési elveken alapul;
- a tervezés során minden részfeladat megoldásánál az összes valószínű alternatívát megvizsgálja;
- a tervezés egész folyamatában optimalizálásokat alkalmaz;
- a hazai és a nemzetközi szállítószalag gyártás Interneten keresztül elérhető adataira támaszkodik, és szükség esetén bővíthető;
- figyelembe veszi a szalagnál előforduló különösebb terheléseket is;
- alkalmas sztochasztikus áramlás esetén történő tervezésre is.

Kutatómunkám során a szakirodalmat az alábbi tématerületekre korlátozva vizsgáltam:

- húzóerődiagram meghatározása,

- szalagtervezésnél alkalmazott szabványok,
- szalagellenállások vizsgálata,
- anyagkeresztmetszet optimális meghatározása,
- a terhelés-változtatás hatása,
- építőelemes, számítógépes szalagtervezés.

A szállítószalag anyagkeresztmetszetének optimális megválasztására már 1975-ben jelent meg kidolgozott elmélet. Eszerint a legnagyobb szekcióterületet a köríves alátámasztás biztosítja. Ezt egy olyan füzérrel lehet biztosítani, ahol nagy a görgőszám. A füzér lehet kötéltengelyű, flexibilis vagy gumitárcsás. Az elmélet kidolgozása során nem vették figyelembe, hogy milyen hatással van az anyagkeresztmetszetre, illetve a szállítóképességre az ömlesztett anyag rézsűszöge, a görgős megtámasztás görgőszáma, a vályúsítás szöge, vagy a hevedersebesség.

A húzóerő közelítő meghatározásának számításával több szabvány is foglalkozik. Csak a külföldieket említve ilyen a német DIN 22101 [34], a szintén német TGL 20-350001 [35], illetve TGL 35378 [36], valamint a nemzetközi szabványok közül az ISO 5048 [37]. Ezek közül a legkorszerűbb, és a legszélesebb körben használt a DIN 22101, így a minimálisan szükséges lefutóerő számításánál ezt javasolt alkalmazni.

A szállítószalag mozgatási ellenállásának meghatározásához léteznek más szerzők által javasolt, továbbfejlesztett módszerek (Behrens [38], Pajer [39], Schwarz [40] és Quass [41]). Végtelen vonóelemmel rendelkező szállítóberendezések vonóellenállás-diagramjának meghatározásával egy- és többhajtásos elrendezésnél vizsgálni kell a hajtás- és feszítéshely megválasztásának hatását a húzóerődiagram jellemzőire. Ezzel kapcsolatosan Alles [42], Cselényi [43] és Köhler [44] ismertetett lehetséges módszereket. Fenti szerzők foglalkoznak speciális, kritikusnak ítélt terhelési esetek elemzésével is, de Cselényi [43] kivételével (a hivatkozott kandidátusi értekezésben a vizsgálat ebből

a szempontból a szállítószalagnál bonyolultabb konvektorokra lett elvégezve), optimáló eljárásokat nem ismertettek.

A húzóerő-diagram jellemzői a szállított anyag fel- és leadási programjától függően sztochasztikus változóként értelmezhetők. Ezzel kapcsolatosan jelentek meg cikkek, de azok gyakorlati eseteket nem vizsgáltak. A modulkategóriák meghatározásához, és az azon helyre kiválasztható modulok tényleges kiválasztásához azonban valós esetek elemzésére is szükség van. Az értekezésemben megfogalmazott elméletnek a szállítószalagokra való alkalmazásához azonban szükség van olyan publikációk megismerésére, amelyek építőelemekből összeállítható szállítószalagok számítógépes tervezésével foglalkoznak. Ilyen a Cselényi, Illés, és Németh szerzők által készített kutatási zárójelentés [45], valamint Eckardt [46] cikke. Ebben bemutatják a tervezés algoritmusát, az építőelemek jellemzőit tartalmazó adatbázist, illetve annak kezelését, valamint módszert adnak az építőelemek kiválasztására. Ugyanakkor a cikk a kiválasztási folyamat alkalmazása során nem alkalmaz optimálási eljárást. Célul fogalmazom meg, hogy a modulkategóriák, illetve elemkategóriák kialakítása során a beruházási költségeket döntően meghatározó összetevők (heveder, görgők, hajtás és feszítés) kiválasztásakor a lehetséges modulokat úgy válasszam ki, hogy fenti költségeket a legalacsonyabban tartsam. Olyan modulokat válasszak ki, amelyek alkalmasak a sztochasztikus hatások kezelésére.

Fentiek figyelembevételével a következő modulkategóriákat alakítottam ki:

- vázszerkezet,
- görgők,
- heveder,
- hajtás,
- feszítés,
- feladóhely,
- leadóhely.

1-es lista

A továbbiakban vizsgálom, hogy az 1-es listában megadott modulkategóriák esetében megadni szükséges értékeket milyen módon lehet meghatározni.

Adott nyomvonalú szállítópálya esetén, ismert szállítandó anyag mellett, előírt szállítási teljesítőképesség biztosításához (ezek azok az elvárt működést biztosító paraméterek, amelyekre az algoritmusnak szüksége van) a következő heveder-keresztmetszeti jellemzőket kell figyelembe venni:

- heveder szélesség,
- görgőszám,
- vályúsítási szög,
- szállítási sebesség,
- keresztmetszet kihasználási tényező.

A szállítási teljesítőképesség meghatározásához használható:

$$Q = 3600 * A_S * v * \rho * \varphi_1 * \varphi_2 \text{ (t/h)}$$

ahol:

A_S – a szalagon lévő szükséges anyagkeresztmetszet (m^2)

v – a heveder sebessége (m/s)

ρ – a szállított anyag sűrűsége (t/m^3)

φ_1 – a szállítópálya maximális emelkedési szögétől függő
teljesítőképességet csökkentő tényező

φ_2 – a szalagra történő feladási módtól függő teljesítőképességet
csökkentő tényező

A φ_1 , φ_2 tényezők a tervezés során megadandó tényezők. Ezek egy konkrét tervezési feladat esetén adottak. A szállítási teljesítőképesség tehát a szalagon lévő szükséges anyagkeresztmetszetnek, a heveder sebességének és a feladási módtól függő teljesítőképességet csökkentő tényezőnek a függvénye. Így felírható:

$$Q = f(A_s, v, \varphi_2)$$

Vizsgálni kell a kialakuló anyagkeresztmetszetet (A_s),

$$A_s = f(B_s, n_G, \lambda, \beta)$$

amihez a következő paramétereket kell megadni:

- B_s - a heveder hasznos szélessége
- n_G - a szalagkeresztmetszetben lévő görgők száma
- λ - a heveder vályúsítási szöge
- β - az anyag szállítása folyamán, a szalagon előálló, a szállított anyagra jellemző rézsűszög

A tervezés során korlátokat kell betartani a hevederszélességre (B), a keresztmetszetben lévő görgőszámra (n_G), a heveder vályúsítási szögére (λ) és a heved sebességére vonatkozóan. Ezek:

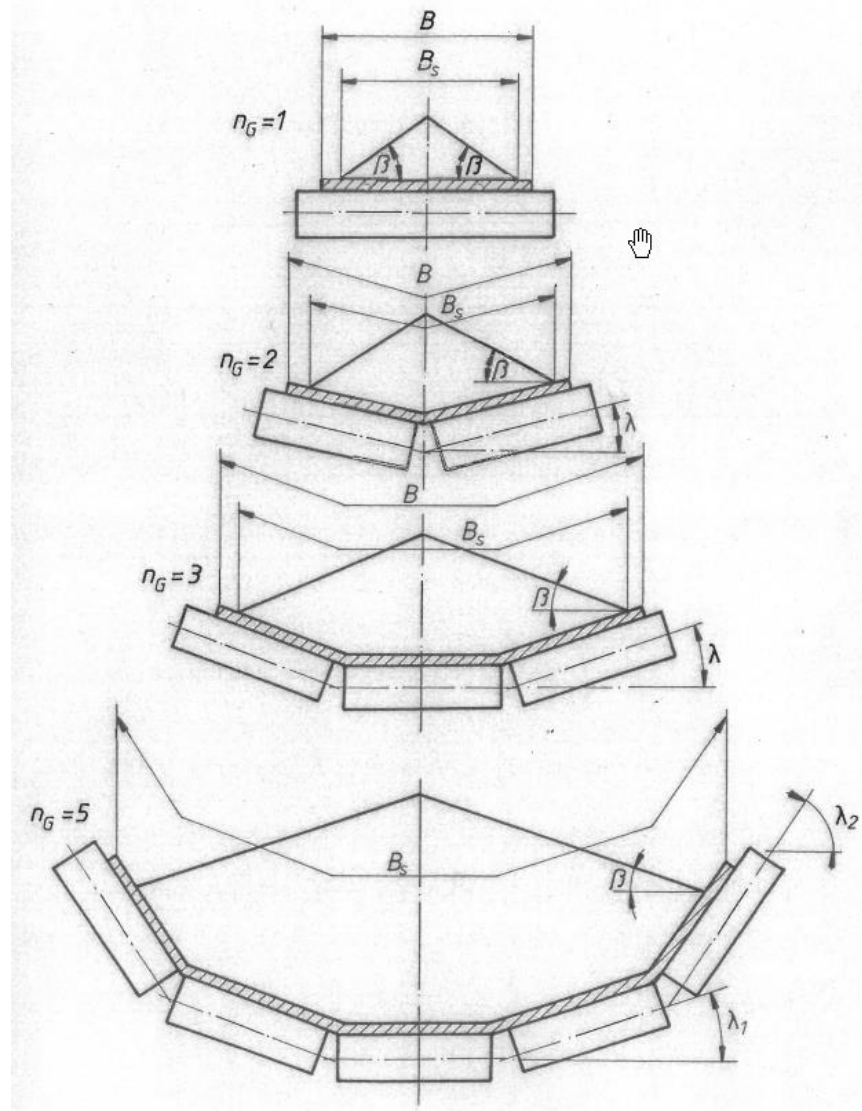
$$B_a \leq B \leq B_f$$

$$n_{Ga} \leq n_G \leq n_{Gf}$$

$$\lambda_a \leq \lambda \leq \lambda_f$$

$$v_a \leq v \leq v_f$$

ahol az a és f indexekkel az alsó- illetve felső korlátok értékére hivatkozok.



25. ábra A szalagon kialakuló anyagkeresztmetszet változatok [47]

A berendezés elvárt teljesítmények megadásakor a modulkategóriák paramétereinek meghatározásához célfüggvényeket kell használni, amelyekkel meghatározható:

- a minimális hevederszélesség (mivel a heveder a berendezés költségének 45%-60%-át teszi ki);
- a szalagkeresztmetszetben lévő minimális görgőszám (a berendezés költségének 15%-20%-a);
- a heveder minimális vályúsítási szöge (minimális érték esetén a heveder élettartama nő);

- a minimális szállítási sebesség (ekkor kisebbek a dinamikus hatások, valamint alacsonyabbak a sebességgel arányos veszteségek, így elegendő lehet kisebb teljesítményű hajtás is);
- a maximális szállítási keresztmetszet kihasználtság;
- a maximális anyag térkitöltési tényező.

A modulkategóriákhoz kiválasztásra kerülő modulok vizsgálatok azokat a modulokat kell kiválasztani, amelyek esetében az elvárt működéshez szükséges lefutóági erő induló értéke megegyezik az adatbázisban már meglévő modulok esetében alkalmazott lefutóági erő induló értékével, vagy értéke meghaladja az elvárt értéket. Szállítószalagok hagyományos tervezése során meghatározó a húzóerődiagram. Amennyiben szükséges, úgy a húzóerődiagram elkészítésével és annak jellemzőinek felhasználásával lehet előállítani az elvárt teljesítményt biztosító paraméterek értékét. Ehhez ismerni kell az induló lefutóági erő nagyságát, amely függ a különböző hajtási módoktól. A húzóerődiagram elkészítése után a következő paraméterek állnak rendelkezésre (Illés [47]):

- az induló lefutóági erő nagysága,
- a görgőosztás a felső heveder-ágon,
- a görgőosztás az alsó heveder-ágon,
- a hajtódob átmérője,
- görgőhossz a felső heveder-ágon,
- görgőhossz az alsó heveder-ágon,
- görgőátmérő értéke a felső heveder-ágon,
- görgőátmérő értéke az alsó heveder-ágon,
- szükséges minimális húzóerő nagysága a hevederben,
- szükséges hajtásszám,
- hajtáshelyen a dobok száma,
- egy dobon lévő motorszám,
- egy dobra eső kerületi erő értéke,
- a hajtódobon alkalmazott átfogási szög,

- a hajtódobon a súrlódási tényező,
- a hajtódobon a tényleges felületi nyomás,
- a felső heveder-ágon lévő görgők forgórész tömegének folyóméterre eső része,
- az alsó heveder-ágon lévő görgők forgórész tömegének folyóméterre eső része,
- a terhelési programok fő ellenállási veszteség tényezői.

A fenti eredmény-paraméterek előállításához meg kell adni:

- a hajtódob üzemállapotát,
- a hajtódob borítását, és
- az alkalmazni kívánt heveder típusát

A szállítózsalagok hagyományos illetve építőelem elvű tervezésének témakörét elemezve, meghatároztam azokat a modul kategóriákat, és az 1-es listában tételesen felsoroltam, amelyek segítségével a szállítózsalag megtervezhető olyan módon, hogy a megadott paraméterek segítségével a szállítózsalag megtervezhető modulokból. A [8] irodalomban ismertetett elv, és az arra épülő, azt továbbfejlesztett, [47]-ben ismertetett algoritmus, és az alapján készített több modulból álló programrendszer ezeket a paramétereket elő tudja állítani, így szükség esetén felhasználható az értekezésemben ismertetett algoritmus bemenő paramétereinek előállítására (a Szakértői Modul részeként) az algoritmusnak gumihevederes szállítózsalagokra való alkalmazása esetében. Ugyanez a több modulból álló programrendszer [47] képes szolgáltatni olyan további paramétereket is, mint a hajtási- és a feszítési helyek és azok jellemzői.

Tézis IV. Kidolgoztam egy új, a szállítózsalagtervezési feladatra alkalmazható mintarendszert. Meghatároztam azokat a modul kategóriákat, amelyekre megadva a berendezés elvárt működését eredményező paraméterek értékét az algoritmus szolgáltatja azokat a modulokat,

amelyekkel biztosítható a berendezés elvárt módon való működése. A mintarendszer a meghatározott modulkategóriákon keresztül alkalmas a már elkészült architektúra modellre és algoritmusra alapozva szállítószalagok esetében történő alkalmazásra. [S19]

6. Az értekezés tézisei

- I. Tézis:** Kidolgoztam egy olyan moduláris rendszerek tervezését támogató rendszermodellt, amely építőelv alapján biztosítja a modulok hatékony illeszkedési vizsgálatát, építve a már korábban elkészített rendszerekből kinyerhető információkra. Az absztrakt berendezések és absztrakt modulok rétegének elemei összerendeződnek a konkrét elemekkel, ahol a kapcsolatok az attribútumokon keresztül szabályokkal felügyeltek. A javasolt modell kiemelkedő rugalmasságot és újrahasznosíthatóságot biztosít. [S1] [S3] [S17]
- II. Tézis:** Kidolgoztam egy új hatékony felhőalapú szoftver/hardver architektúra modellt az 1. tézisben megadott elvű tervezési modell implementálására, így biztosítva a felhőalapú szolgáltatások előnyeit, mint platform függetlenség, hatékony szoftverfejlesztés/verziókövetés, helyfüggetlenség, hatékonyság. [S2] [S7-S9] [S18]
- III. Tézis:** A modulok kiválasztására kidolgoztam egy új algoritmust, amely a kijelölt célfüggvényhez meghatározza az optimális berendezés-struktúrát. Az algoritmus több optimalizálási fázist tartalmaz az egyes rétegek elemeinek az összerendeléséhez. Az eljárás során meghatározásra kerülnek az optimális paraméter intervallumok, kompatibilitási feltételek. A megadott feltételek által meghatározott tartományon történik a kijelölt célfüggvény szerinti optimum meghatározása. [S17] [S19]

IV.Tézis: Kidolgoztam egy új, a szállítószalagtervezési feladatra alkalmazható mintarendszert. Meghatároztam azokat a modulkategóriákat, amelyekre megadva a berendezés elvárt működését eredményező paraméterek értékét az algoritmus szolgáltatja azokat a modulokat, amelyekkel biztosítható a berendezés elvárt módon való működése. A mintarendszer a meghatározott modulkategóriákon keresztül alkalmas a már elkészült architektúra modellre és algoritmusra alapozva szállítószalagok esetében történő alkalmazásra. [S19]

7. Theses of the dissertation

- I. **Thesis:** I have developed a system model that supports the design of modular systems, which, based on the construction principle, ensures the efficient fit testing of the modules, based on the information that can be extracted from the systems that have already been prepared. The elements of the layer of abstract equipment and abstract components are arranged with the concrete elements, where the relationships are supervised by rules through the attributes. The proposed model provides outstanding flexibility and reusability. [S1] [S3] [S17]

- II. **Thesis:** I developed a new efficient cloud-based software/hardware architecture model to implement the principled design model given in thesis 1, thus ensuring the advantages of cloud-based services, such as platform independence, efficient software development/version tracking, location independence, efficiency. [S2] [S7-S9] [S18]

- III. **Thesis:** I developed a new algorithm for the selection of modules, which determines the optimal equipment structure for the selected objective function. The algorithm contains several optimization phases for matching the elements of each layer. The optimal parameter intervals and compatibility conditions are determined during the procedure. The optimum is determined according to the designated objective function in the range defined by the specified conditions. [S17] [S19]

IV. Thesis: I developed a new prototype system applicable to the conveyor belt design task. I have defined the module categories for which, given the value of the parameters resulting in the expected operation of the equipment, the algorithm select the modules that can be used to ensure the operation of the equipment in the expected manner. The prototype system can be used based on the already completed architecture model and algorithm in the case of conveyor belts that through specific module categories. **[S19]**

8. Összefoglalás

Értekezésemben az építőelemes berendezések felhőalapú számítógépes tervezésének témakörével foglalkoztam. Ez a terület két szempontból is hangsúlyos. Az első szempont a címben is megjelenő felhőalapú tervezés, azon belül is a felhőszolgáltatások, amelyek a mindennapi életben is egyre szélesebb körben kerültek bevezetésre az informatika szinte minden területén. A másik szempont, amely szintén megjelenik az értekezés címében, az építőelemes tervezés. Ez nem kimondottan új terület, viszont létezik olyan aspektusa (a modulkategóriák, az elemkategóriák, és az azokhoz kapcsolódó új fogalmak bevezetésével és azok használatával), amelyben nagy lehetőségek rejlenek.

Értekezésemben részletesen, több publikációs adatbázisra támaszkodva kutattam a releváns szakirodalmat. Ennek érdekében a keresés során több kulcsszó kombinációt használtam az elmúlt közel tíz év cikkeit megvizsgálva. A kulcsszavakat az értekezés témájához illesztve határoztam meg. A felhőalapú logisztikai tervezéssel foglalkozó, azokra épülő kutatásokhoz köthető publikációk száma folyamatosan nőtt. Hasonlóan, de nem ilyen mértékben nőtt a felhőalapú moduláris tervezéssel foglalkozó tudományos cikkek száma is. A megjelent publikációk részletes vizsgálata azonban azt mutatta, hogy a találatok false találatok. A két területet összekapcsolva (*felhőalapú logisztikai moduláris tervezés*) nem jelent meg olyan cikk, amely ezt a területet kutatta volna. Ez azt jelzi, hogy ezen a területen jelentős kutatási potenciál van.

Az értekezésben bemutattam a felhőtechnológiák kialakulását, az abban rejlő lehetőségeket. Kitértem a felhőszolgáltatások megjelenésére, egymásra épülésére, az elterjedt felhő architektúrára.

Az irodalomkutatás során nem volt fellelhető a témához szorosabban illeszkedő tudományos cikk, de jelent meg egy olyan publikáció, amely felhőalapú gyártástervezés eredményeit mutatta be. Ezt részletesebben is bemutattam. A szerzők által publikált eredmények azt mutatták, hogy minél összetettebb egy adott

tervezési feladat, annál hatékonyabb a felhőtechnológia alkalmazása. Ezzel igazolódott az a hipotézisem, hogy az architektúrának felhőtechnológiára kell épülnie.

A moduláris tervezésre összpontosítva elkészítettem azt a modellt, amely lehetővé teszi a berendezést alkotó modulok hatékony illeszkedés vizsgálatát. Ez egyben az első tézisem is. Ilyen modellt a szakirodalom elemzése során nem találtam.

A következő részben kidolgoztam egy olyan felhőalapú szoftver/hardver architektúrát, amely alkalmas az első tézisben kialakított modell implementálására. A kidolgozott architektúra a lehető legnagyobb mértékben támaszkodik a felhőszolgáltatásokra, lehetőséget adva a felhő által biztosított hatékonyságnövelő eszközök használatára. Ez egyben a második tézisem. Ilyen architektúrát a szakirodalom elemzése során nem találtam.

Az értekezés következő részében a kidolgozott architektúrára építve elkészítettem azt az algoritmust, amely elvégzi egy tetszőleges berendezés építőelemekre épülő tervezését. Ehhez új fogalmakat vezettem be. Az algoritmus kezeli azokat az eseteket, amikor a berendezés tervezéséhez nem sikerült az adatbázisban vagy megfelelő modult, vagy megfelelő elemet találni. Ez egyben a harmadik tézisem.

Az értekezés utolsó részében kidolgoztam egy új, a szállítószalagtervezési feladatra alkalmazható mintarendszert. Meghatároztam azokat a modulkategóriákat, amelyekre megadva a berendezés elvárt működését eredményező paraméterek értékét az algoritmus szolgáltatja azokat a modulokat, amelyekkel biztosítható a berendezés elvárt módon való működése. A mintarendszer a meghatározott modulkategóriákon keresztül alkalmas a már elkészült architektúra modellre és algoritmusra alapozva szállítószalagok esetében történő alkalmazásra. Ez egyben a negyedik tézisem.

9. Summary

In my dissertation, I dealt with the topic of cloud-based computer design of building element equipment. This area is important from two points of view. The first aspect is cloud-based planning, which also appears in the title, including cloud services, which have been increasingly introduced in everyday life in almost all areas of IT. The other aspect, which also appears in the title of the thesis, is building element design. This is not exactly a new field, but there is an aspect of it that has great potential.

In my dissertation, I researched the relevant literature in detail, relying on several publication databases. To do this, I used several combinations of keywords in the search, examining the articles of the last ten years. I defined the keywords according to the topic of the thesis. The number of publications dealing with cloud-based logistics planning and linked to research based on them has been constantly increasing. Similarly, but not to the same extent, is the number of scientific articles dealing with cloud-based modular design. However, a detailed examination of the published publications showed that the results were false results. Connecting the two areas (*cloud-based logistics modular planning*), no articles were published that researched this area. This indicates that there is significant research potential in this area.

In the dissertation, I presented the development of cloud technologies and the possibilities inherent in them. I touched on the appearance of cloud services, their building on each other, and the widespread cloud architecture.

The literature search did not find a scientific article more closely related to the topic, but a publication was published that presented the results of cloud-based production planning. I presented this in more detail. Their detailed results showed that the more complex a given design task is, the more effective the application of cloud technology is. This confirmed my hypothesis that the architecture should be based on cloud technology.

Focusing on modular design, I created its model, which is also my first thesis. The model makes it possible to test the effective fit of the modules that make up the equipment. I did not find such a model during the analysis of the literature.

In the next part, I developed a cloud-based software/hardware architecture that is suitable for implementing the model developed in the first thesis. The developed architecture relies on cloud services as much as possible, giving the opportunity to use the efficiency-enhancing tools provided by the cloud. This is also my second thesis. I did not find such an architecture during the analysis of the literature.

In the next part of the dissertation, based on the elaborated architecture, I created the algorithm that performs the design of any device based on building elements. For this, I introduced new concepts. The algorithm handles those cases when it was not possible to find either a suitable module or a suitable element in the database for the design of the equipment. This is also my third thesis.

In the last part of the dissertation, I developed a new prototype system which can apply to the conveyor belt design task. I have defined the module categories for which, given the value of the parameters resulting in the expected operation of the equipment, the algorithm selects the modules that can be used to ensure the operation of the equipment in the expected manner. The prototype system can be used based on the already completed architecture model and algorithm in the case of conveyor belts that through specific module categories. This is also my fourth thesis.

10. Irodalomjegyzék

- [1] Feng, Chen; Xiao, Yong; Willette, Aaron; McGee, Wes; Kamat, Vineet R.: Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites (Automation in Construction, 2015, DOI10.1016/j.autcon.2015.06.002)
- [2] Nag, Sreeja; Li, Alan S.; Merrick, James H.: Scheduling algorithms for rapid imaging using agile Cubesat constellations (Advances in Space Research, 2018, DOI10.1016/j.asr.2017.11.010)
- [3] Rausch, Christopher; Nahangi, Mohammad; Perreault, Melanie; Haas, Carl T.; West, Jeffrey: Optimum Assembly Planning for Modular Construction Components (Journal of Computing in Civil Engineering, 2017, DOI10.1061/(ASCE)CP.1943-5487.0000605)
- [4] Kretsis, Aristotelis; Christodoulopoulos, Konstantinos; Kokkinos, Panagiotis; Varvarigos, Emmanouel: Planning and Operating Flexible Optical Networks: Algorithmic Issues and Tools (IEEE Communications Magazine, 2014, DOI10.1109/MCOM.2014.6710065)
- [5] Saah, David; Tenneson, Karis; Matin, Mir; Uddin, Kabir; Cutter, Peter; Poortinga, Ate; Nguyen, Quyen H.; Patterson, Matthew; Johnson, Gary; Markert, Kel; Flores, Africa; Anderson, Eric; Weigel, Amanda; Ellenberg, Walter L.; Bhargava, Radhika; Aekakkararunroj, Aekkapol; Bhandari, Biplov; Khanal, Nishanta; Housman, Ian W.; Potapov, Peter; Tyukavina, Alexandra; Maus, Paul; Ganz, David; Clinton, Nicholas; Chishtie, Farrukh: Land Cover Mapping in Data Scarce Environments: Challenges and Opportunities (Frontiers in Environmental Science, 2019, DOI10.3389/fenvs.2019.00150)

- [6] Borraz, Raul; Navarro, Pedro J.; Fernandez, Carlos; Maria Alcover, Pedro: Cloud Incubator Car: A Reliable Platform for Autonomous Driving (Applied Sciences-Basel, 2018, DOI10.3390/app8020303)
- [7] Zhang, Xianyu; Ming, Xinguo; Liu, Zhiwen; Qu, Yuanju; Yin, Dao: General reference model and overall frameworks for green manufacturing (Journal of Cleaner Production, 2019, DOI10.1016/j.jclepro.2019.117757)
- [8] Borges Oliveira, Dario Augusto; Leal-Taixe, Laura; Feitosa, Raul Queiroz; Rosenhahn, Bodo: Automatic tracking of vessel-like structures from a single starting point (Computerized Medical Imaging and Graphics, 2015, DOI10.1016/j.compmedimag.2015.11.002)
- [9] Cheikhrouhou, Omar; Koubaa, Anis; Zarrad, Anis: A Cloud Based Disaster Management System (Journal of Sensor and Actuator Networks, 2020, DOI10.3390/jsan9010006)
- [10] Valizadeh, Siavash; Valilai, Omid Fatahi; Houshmand, Mahmoud; Vasegh, Zahra: A novel digital dentistry platform based on cloud manufacturing paradigm (International Journal of Computer Integrated Manufacturing, 2019, DOI10.1080/0951192X.2019.1686170)
- [11] Swati I. Bairagi, Ankur O. Bang: Cloud Computing: History, Architecture, Security Issues (International Journal of Advent Research in Computer and Electronics (IJARCE), 2015, E-ISSN: 2348-5523)
- [12] Ari Liberman Garcia: The Evolution of the Cloud (The Work, Progress and Outlook of Cloud Infrastructure) (Massachusetts Institute of Technology, 2015, <https://dspace.mit.edu/bitstream/handle/1721.1/100311/932065967-MIT.pdf;sequence=1>)
- [13] Wei, Wei; Zhou, Feng; Liang, Feng-Fei: Product platform architecture for cloud manufacturing (Advanced Manufacturing, 2020, <https://doi.org/10.1007/s40436-020-00306-1>)

- [14] Guide to Cloud Computing Architectures (Packt Publishing, <https://www.networkcomputing.com/cloud-infrastructure/guide-cloud-computing-architectures>, 2018)
- [15] The Cloud Computing reference model (<https://cloudmanfr.wordpress.com/2017/10/31/the-cloud-computing-reference-model>, 2117)
- [16] Aldossary, Mohammad: A Review of Energy-Related Cost Issues and Prediction Models in Cloud Computing Environments (Computer Systems Science & Engineering, 2021, DOI:10.32604/csse2021.014974)
- [17] Priya Pedemkar: Fog Computing Architecture (<https://www.educba.com/fog-computing-architecture/>)
- [18] Cloud Computing – A Five Layer Model (Cloud Computing – A Five Layer Model (intervision.com, 2021)
- [19] Givehchi, Omid; Jasperneite, Juergen: Industrial Automation Services as part of the Cloud: First Experiences (ResearchGate, <https://www.researchgate.net/publication/257402460>, 2013)
- [20] Khajehei, Karnyab: Green Cloud and Virtual Machines Migration Challenges (Indian Journal of Science and Technology, DOI: 10.17485/ijst/2016/v9i5/71386, 2016)
- [21] Banica, Logica; Burtescu, Emil: Advanced Security Models for Cloud Infrastructures (ResearchGate, <https://www.researchgate.net/publication/266077953>, 2014)
- [22] IBM: Cloud Computing History (<https://www.ibm.com/cloud/blog/cloud-computing-history>)
- [23] InterVision: Cloud Computing – A Five Layer Model (<https://intervision.com/cloud-computing-five-layer-model/>)

- [24] Sedani, Jatankumar; Doshi, Minal: Cloud Computing: From the Era of Beginning to Present (International Journal of Novel Research in Computer Science and Software Engineering, 2015, ISSN 2394-7314)
- [25] Wei Y, Blake MB: Service-oriented computing and cloud computing: challenges and opportunities. (IEEE Conference on Internet Computing, pp. 72–75, 2010)
- [26] Singh, B; Dhawan, S; Arora, A: A view of cloud computing (International Conference on Computing Technology, pp. 50–58, 2013)
- [27] Ding, B; Yu Sun, LJ: A cloud-based collaborative manufacturing resource sharing services (International Conference on Information Technology, pp. 1258–1264, 2012)
- [28] Ren, L; Zhang, L; Wang, L. et al: Cloud manufacturing: key characteristics and applications (International Conference on Computing Integrated Manufacturing, pp. 501–515, 2017)
- [29] Zhang, L; Luo, YL; Fan, WH, et al: Analyses of cloud manufacturing and related advanced manufacturing models (International Conference on Computing Integrated Manufacturing System, pp. 458–468, 2011)
- [30] Xu, X: From cloud computing to cloud manufacturing (Conference on Robots and Computer Integrated Manufacturing 28, pp. 75–86, 2012)
- [31] Ren, L;, Zhang, L; Zhang, YB, et al: Resource virtualization in cloud manufacturing. (Conference on Computer Integrated Manufacturing System 17, pp. 511–518, 2011)
- [32] Wei, W; Liang, PF: A product platform architecture for cloud manufacturing (Proceedings of 2018 48th international conference on computers and industrial engineering, Auckland, New Zealand, 2018)

- [33] Capacity: History of Cloud Storage (<https://capacity.com/cloud-storage/history-of-cloud-storage/>)
- [34] DIN 22101: Continuous conveyor – Belt conveyors for loose bulk materials; Basics for calculation and dimensioning (<https://pdfcoffee.com/qdownload/din-22101-pdf-free.html>)
- [35] TGL 20-350001: Conveyors for continuous mechanical handling; Belt conveyors calculation method (<https://bauarchivddr.bbr-server.de/bauarchivddr/archiv/tglarchiv/tgl20-1bis20-x/tgl-20-350001-mrz-1965.pdf>)
- [36] TGL 35378: Continuous conveyors; Belt conveyors – Basis of calculation (<https://bauarchivddr.bbr-server.de/bauarchivddr/archiv/tglarchiv/tgl30001bis40000/tgl35001bis35500/tgl-35378-apr-1981.pdf>)
- [37] Continuous mechanical handling equipment – Belt conveyors with carrying idlers – Calculation of operating power and tensile forces (<https://cdn.standards.iteh.ai/samples/11069/db7ed9c359514146bdbb893dc468002d/ISO-5048-1989.pdf>)
- [38] Behrens, U.: Untersuchungen zum Walkwiderstand schwerer Förderbandanlagen. (Dissertation, Technische Hochschule Hannover, 1967.)
- [39] Pajer, G.: Zum Verformungswiderstand des Fördergutstroms auf Gurtförderern (Hebezeuge und Fördermittel, 20.k. 8. sz., p228-231)
- [40] Schwarz, F.: Zum Eindrückrollenwiderstand zwischen Fördergut und Tragrolle (Fördern und Heben, 17. H.12)
- [41] Quass, H.: Betrachtung zur Berechnung des Bewegungswiderstandes an Gurtbandförderern (Bergbautechnik, 12., p 650-656)
- [42] Alles, R.: Fördergurt Berechnung (Continental Gummiwerke A. G., Hannover (2))

- [43] Cselényi, J.: Végetlen vonóelemes anyagmozgató berendezések méretezésének alapjai állandósult üzemállapot esetén (Kandidátusi értekezés, Miskolc. 1978, p 1-130)
- [44] Köhler, W.: Zur Berechnung und Dimensionierung von Gurtbandförderern (Wissenschaftliche Zeitschrift der Technischen Hochschule Otto von Guericke, 18. 6/7, 1974, p 675-679)
- [45] Cselényi, J., Illés, B., Németh, J.: Építőelemekből felépíthető szállítószalagok számítógépes tervezése (NME Szállítóberendezések Tanszéke, Kutatási zárójelentés, 1980)
- [46] Eckardt, G., Köhler, W.: Automatisierte Variantenkonstruktion mit Hilfe elektronischer Datenverarbeitungsanlagen (Hebezeuge und Fördermittel, 1971, p 1-5)
- [47] Illés, B.: Gumihevederes szállítószalagok interaktív tervezése moduláris programrendszerrel (egyetemi doktori értekezés, NME, Miskolc, 1986, p 1-183)

11. Saját publikációk

- [S1] **Wagner, György**; Kovács, László; Illés, Béla: Matematikai módszerek a logisztika területén (XXX. Nemzetközi Gépészeti Konferencia, 2022, <https://ojs.emt.ro/oget/article/view/863/811>)
- [S2] **Wagner, György**; Kovács, László; Illés, Béla: Moduláris elemekből felépíthető logisztikai berendezések tervezése (XXX. Nemzetközi Gépészeti Konferencia, 2022, <https://ojs.emt.ro/oget/article/view/864/8112>)
- [S3] Fűkő, László; Illés, Béla; **Wagner, György**: A hagyományos és a rugalmas gyártórendszerek logisztikai aspektusai (Multidiszciplináris Tudományok: A Miskolci Egyetem Közleménye 10, 2020, <https://doi.org/10.35925/j.multi.2020.3.45>)
- [S4] Illés, Béla; Skapinyecz, Róbert; **Wagner, György**: Description of a Method for the Handling of Customer Needs in Logistic (Lecture Notes in Mechanical Engineering F12, 2017, https://doi.org/10.1007/978-3-319-51189-4_31)
- [S5] Illés, Béla; Skapinyecz, Róbert; **Wagner, György**; Glistau, Elke; Machado, Coello: Application of QFD for the Handling of Customer Needs in Automotive Industry (IX. International Scientific Conference of Mechanical Engineering, 2016)
- [S6] **Wagner, György**: Security Problems of Engine Control Software (XXIII. Nemzetközi Gépészeti Konferencia, pp. 431-434, 2015)
- [S7] **Wagner, György**: Security of Cloud Based Storage (XXI. Nemzetközi Gépészeti Konferencia, pp. 443-446, 2013)
- [S8] **Wagner, György**: On-line tárolóhelyek biztonsága (GÉP 64, pp. 80-83, 2013)

- [S9] **Wagner, György:** Logisztikai rendszerek informatikai architektúrája (TÁMOP-4.1.2.-08/1/A-2009-0001 G2, pp. 1-27, 2011)
- [S10] **Wagner, György:** Virtuális Logisztikai Központ által kezelendő információk (TÁMOP-4.2.1.08/1-2008-0006 PP3, pp. 1-24, 2011)
- [S11] **Wagner, György:** VIR információs technológia alapjai (TÁMOP-4.1.2.-08/1/A-2009-0049, pp. 1-28, 2011)
- [S12] **Wagner, György;** Cselényi, József; Csizmadia, László: Konvektorok számítógépes tervezése sztochasztikus terhelések esetén (MicroCAD 90, paper 12, 1990)
- [S13] Cselényi, József; Mang, Béla; **Wagner, György:** Konvektor szállítóberendezések tervezésére szolgáló programrendszer IBM PC számítógépre (GÉP 41, pp. 244-249, 1989)
- [S14] **Wagner, György;** Cselényi, József; Mang, Béla: Függőspályás anyagmozgató rendszerek számítógépes tervezése (Üzemi anyagmozgatási rendszerek számítógépes tervezése szimpózium, 1988)
- [S15] **Wagner, György;** Cselényi, József; Csizmadia, László; Mang, Béla: Rechner unterstützte Planung der Robotisierte Bedienung einer feuerfeste Ziegel Erzeugenden Presse (Konferenz zur Materialhandhabung, Belgrád, pp. 19, 1988)
- [S16] **Wagner, György;** Csekő, Béla; Mang, Béla; Központi tárolóval rendelkező görgőspályás rugalmas anyagmozgató rendszer szimulációs vizsgálata (XXIV. Borsodi Műszaki és Közgazdasági Hetek, 1986)
- [S17] Kovács, László; **Wagner, György:** Heurisztikus módszer a modulok összerendelésének optimalizálására (Production Systems and Information Engineering, 2023)
- [S18] **Wagner, György;** Kovács, László: Felhőalapú moduláris tervező program architektúrája (Production Systems and Information Engineering, 2023)

[S19] **Wagner, György**; Kovács, László: Felhőalapú moduláris tervezés algoritmusai (Production Systems and Information Engineering, 2023)

I. Ábrajegyzék

1. ábra Publikációk számának százalékos értéke évek szerint az (1) keresési feltételnek megfelelően.....	20
2. ábra Publikációk számának százalékos értéke évek szerint a (2) keresési feltételnek megfelelően.....	21
3. ábra Publikációk számának százalékos értéke évek szerint a (3) keresési feltételnek megfelelően.....	22
4. ábra Publikációk számának százalékos értéke évek szerint a (4) keresési feltételnek megfelelően.....	24
5. ábra Publikációk számának százalékos értéke évek szerint az (5) keresési feltételnek megfelelően.....	25
6. ábra Publikációk által lefedett tudományterületek megoszlása a Scopus adatbázisa szerint.....	26
7. ábra Publikációk által lefedett tudományterületek megoszlása a Web of Science adatbázisa szerint.....	27
8. ábra Publikációk által lefedett tudományterületek megoszlása a Science Direct adatbázisa szerint.....	28
9. ábra Publikációk darabszámának alakulása a Web of Science adatbázisa alapján.....	29
10. ábra Hivatkozások számának alakulása a Web of Science adatbázisa alapján.....	30
11. ábra A virtualizációs réteg szolgáltatásai.....	36
12. ábra Felhőszolgáltatási modellek egymásra épülése.....	40
13. ábra Az építőelem elvű (moduláris) tervezés modellje.....	47
14. ábra A kidolgozott felhőarchitektúra.....	51
15. ábra A kidolgozott felhőarchitektúra kliens rétege.....	53
16. ábra A kidolgozott felhőarchitektúra SaaS rétege.....	56
17. ábra A kidolgozott felhőarchitektúra PaaS rétege.....	56

18.	ábra A kidolgozott felhőarchitektúra IaaS rétege	57
19.	ábra A modulkategóriákat kezelő algoritmus	60
20.	ábra A berendezés elvárt működését biztosító algoritmus	61
21.	ábra, Lehetséges összerendelések költség hisztogramja.....	68
22.	ábra, Lehetséges összerendelések költség hisztogramja.....	68
23.	ábra, Globális (x-koordináta) és lokális (y-koordináta) optimumok összevetése.....	69
24.	ábra, Az optimalizációs és teljes leszámlálás időkölségei.....	70

II. Táblázatok jegyzéke

1.	táblázat Tudományos publikációk megoszlása a Scopus esetében .	17
3.	táblázat Tudományos publikációk megoszlása a Web of Science esetében.....	18
4.	táblázat Tudományos publikációk megoszlása a Science Direct esetében.....	18
5.	táblázat Tudományos publikációk darabszámának %-os megoszlása az (1) keresési feltételnek megfelelően.....	19
6.	táblázat Tudományos publikációk darabszámának %-os megoszlása a (2) keresési feltételnek megfelelően	21
7.	táblázat Tudományos publikációk darabszámának %-os megoszlása a (3) keresési feltételnek megfelelően	22
8.	táblázat Tudományos publikációk darabszámának %-os megoszlása a (4) keresési feltételnek megfelelően	23
9.	táblázat Tudományos publikációk darabszámának %-os megoszlása az (5) keresési feltételnek megfelelően.....	24
10.	táblázat A kiválasztott tudományterületeken készült publikációkra való hivatkozások száma a Web of Science alapján.....	31