



Universitatea
Transilvania
din Braşov

ŞCOALA DOCTORALĂ INTERDISCIPLINARĂ

Facultatea de Inginerie Electrică și Știința Calculatoarelor

George Alex STELEA

**Contribuții la Managementul prin Servicii Web
în Internetul Obiectelor
Contributions to Internet of Things Management
through Web Services**

REZUMAT / ABSTRACT

**Conducător științific
Prof.univ.dr.ing. Florin SANDU**

BRAȘOV, 2019

D-lui (D-nei)

COMPONENTA

Comisiei de doctorat

Numită prin ordinul Rectorului Universității Transilvania din Brașov

Nr. din

PREȘEDINTE: Prof.univ.dr.ing. Laurențiu Mihail IVANOVICI
Universitatea "Transilvania" din Brașov

CONDUCĂTOR ȘTIINȚIFIC: Prof.univ.dr.ing. Florin SANDU
Universitatea "Transilvania" din Brașov

REFERENȚI: Prof.univ.dr.ing. Adriana SÎRBU
Universitatea Tehnică „Gh. Asachi” din Iași

Prof.univ.dr.ing. Tudor Petru PALADE
Universitatea Tehnică din Cluj-Napoca

Prof.univ.dr.ing. Petre Lucian OGRUȚAN
Universitatea "Transilvania" din Brașov

Data, ora și locul susținerii publice a tezei de doctorat:

21 septembrie, ora 10, sala N.2.1

Eventualele aprecieri sau observații asupra conținutului lucrării vă rugăm să le transmiteți în timp util, pe adresa george.stelea@unitbv.ro

Totodată vă invităm să luați parte la ședința publică de susținere a tezei de doctorat.

Vă mulțumim.

Cuprins

	Pg.teză	Pg.rezumat
Introducere	12	8
1 Stadiul actual în Internetul Obiectelor (IoT)	20	16
1.1. IoT - Caracteristici cheie	20	16
1.1.1. Taxonomie	20	16
1.1.2. Avantajele IoT	24	17
1.1.3. Tendințe de evoluție tehnologică	25	18
1.2. Arhitectura IoT	26	18
1.2.1. Complexitate	27	19
1.2.2. Fog Computing	28	19
1.2.3. Cadre de lucru (Frameworks)	28	20
1.2.4. Tehnologii pentru IoT	29	20
1.2.5. Confidențialitate și protecția identității	30	20
1.2.6. Aspecte de securitate	31	20
1.3. Internetul obiectelor multimedia (IoMT)	32	21
1.3.1. Concept	32	21
1.3.2. Rolul și caracteristicile principale ale IoMT	34	21
1.3.3. Domeniul de aplicabilitate și standardizare a IoMT	35	22
1.3.4. Provocări în dezvoltarea IoMT	38	22
1.4. Sumarul capitolului	39	23
2 Servicii, protocoale și conectivitate în IoT	40	24
2.1. Orientarea pe servicii în IoT	40	24
2.1.1. Arhitectura orientată pe Servicii (SOA)	40	24
2.1.2. REST - Representational state transfer	43	25
2.1.3. Microservicii	46	25
2.2. Arhitectura bazată pe evenimente	48	26
2.2.1. Concept și model arhitectural	48	26
2.2.2. Arhitectura orientată pe servicii condusă de evenimente	51	26
2.3. Protocoale de comunicații pentru IoT	51	27
2.3.1. HTTP (Hypertext Transfer Protocol)	53	27
2.3.2. DDS (Data Distribution Service)	54	27
2.3.3. AMQP (Advanced Message Queueing Protocol)	55	28
2.3.4. CoAP (Constrained Application Protocol)	55	28
2.3.5. MQTT (Message Queue Telemetry Transport Protocol)	56	28
2.3.6. XMPP (Extensible Messaging and Presence Protocol)	57	28
2.4. Conectarea senzorilor la Internet și Web	58	29
2.4.1. Integrarea senzorilor	58	29
2.4.2. Modelarea datelor de la senzori	58	29
2.4.3. Rețele de senzori	59	29
2.5. Contribuție la analiza de date în timp real folosind servicii și metadata web	60	30
2.5.1. Metadata și marcare semantică dinamică	60	30

2.5.2.	Servicii web și localizarea procesărilor aferente	62	32
2.5.3.	Statistici și analize de date în timp real	64	32
2.6.	Sumarul capitolului	66	34
3	Abordarea IoT ca Rețea Socială	67	35
3.1.	Social IoT	67	35
3.1.1.	Concept și descrierea sistemului	67	35
3.1.2.	Cum pot legăturile sociale îmbunătăți comunicațiile IoT	68	36
3.1.3.	Descoperirea releelor și selecția nodurilor de tip peer	69	36
3.1.4.	Selectarea modului de comunicare	71	37
3.1.5.	Alocarea și gestionarea resurselor	73	38
3.2.	Web of Things (WoT)	74	38
3.2.1.	Concept	74	38
3.2.2.	Arhitectură	75	38
3.2.3.	Metode și tehnologii folosite	77	39
3.2.4.	Amenințările de securitate din Web of Things	78	39
3.3.	Metodă de compoziție dinamică a serviciilor colaborative în IoT	79	40
3.3.1.	Demonstrator SIoT - Platformă IoT pentru integrarea și testarea serviciilor colaborative	79	40
3.3.2.	Arhitectură	80	40
3.3.3.	Nivelul aplicație	82	41
3.4.	Sumarul capitolului	86	44
4	Integrarea IoT în Cloud prin comunicații web de timp real (Web-RTC)	87	45
4.1.	Comunicații web de timp real	87	45
4.1.1.	Arhitectura WebRTC	87	45
4.1.2.	API WebRTC	88	46
4.1.3.	Protocoalele STUN și TURN	92	47
4.1.4.	Semnalizarea în WebRTC	94	47
4.1.5.	Avantaje și motive pentru a folosi tehnologia WebRTC	94	48
4.1.6.	Biblioteci și seturi de instrumente WebRTC	95	48
4.2.	Contribuție la comunicarea peer-to-peer în timp real	96	49
4.2.1.	Mediul de dezvoltare și mediul de control al apelurilor	96	49
4.2.2.	Gestionarea conexiunii	98	50
4.2.3.	Gestionarea fluxului	99	51
4.2.4.	Interfața grafică cu utilizatorul	99	51
4.3.	Contribuție la autentificarea în Cloud prin procesarea obiectelor multimedia	101	52
4.3.1.	Tehnologii și implementare	101	52
4.3.2.	Abordarea fluxului de lucru pe rețele neuronale	101	52
4.3.3.	Abordarea prin servicii integrate Cloud	104	53
4.4.	Sumarul capitolului	108	54
5	Servicii Web pentru Telematică în IoT	109	55
5.1.	Contribuții la tele-măsurare folosind instrumentație virtuală și servicii web	109	55
5.1.1.	Transmiterea datelor de la senzori și traductoare inteligente	110	56
5.1.2.	Serverul de instrumentație Cloud și tele-măsurarea bazată pe TCP	111	57
5.1.3.	Caracteristicile cheie ale soluției în contextul Big Data	115	57

5.1.4.	Panoul de control bazat și publicat pe web	116	58
5.1.5.	Implementarea serviciilor web de prezentare	117	58
5.2.	Contribuție la accesul receptiv la terminal în IoT	120	59
5.2.1.	Concept	120	59
5.2.2.	Reactivitate și receptivitate	121	60
5.2.3.	Panoul "CON-INTEL Dashboard"	125	61
5.3.	Sumarul capitolului	130	62
6	Dezvoltare Web și managementul semantic pentru IoT	131	63
6.1.	Semantică și ontologii	131	63
6.1.1.	Web-ul Semantic	131	63
6.1.2.	Soluții Web semantice	132	63
6.1.3.	Provocări ale Web-ului Semantic	132	63
6.2.	Standarde și taxonomii	133	64
6.2.1.	Resource Description Framework	134	64
6.2.2.	RDFa (Resource Description Framework in Attributes)	135	64
6.2.3.	Microformat	138	65
6.2.4.	Microdata	139	65
6.2.5.	JSON-LD (JavaScript Object Notation for Linked Data)	140	65
6.3.	Web Semantic al obiectelor	141	66
6.3.1.	Rețele semantice de senzori	142	66
6.3.2.	Modelarea și adnotarea semantică a datelor senzorilor	143	66
6.3.3.	Rețelele senzorilor semantici W3C	143	67
6.4.	Contribuție la accesibilitatea și managementul interfețelor semantice în IoT	145	67
6.4.1.	Aplicații Rich Internet Accesibile	146	68
6.4.2.	AMP - Accelerated Mobile Pages	150	69
6.5.	Contribuție la managementul resurselor utilizând capabilități offline progresive și inter-platformă	152	70
6.5.1.	Adaptarea conținutului la terminal	152	70
6.5.2.	Management prin metodologia web progresivă	154	71
6.5.3.	Capabilități offline	156	71
6.6.	Sumarul capitolului	159	73
7	Concluzii generale și contribuții originale	161	75
7.1.	Principalele contribuții originale	162	76
7.2.	Direcții viitoare de cercetare	167	81
7.3.	Validarea și diseminarea rezultatelor științifice în publicații și proiecte de cercetare	168	82
7.3.1.	Lucrări proprii indexate ISI	168	82
7.3.2.	Lucrări proprii în curs de indexare ISI	170	83
7.3.3.	Lucrări proprii indexate în alte BDI (EBSCO)	170	83
7.3.4.	Monografii și capitole din cărți cu caracter științific	171	84
7.3.5.	Proiecte de cercetare	171	84
	Bibliografie	171	85
	Rezumat (lb. română/lb. engleză)	xlvi	i
	Curriculum vitae (lb. română/lb. engleză)	xlvii	iii

Table of contents

	Pg.thesis	Pg.abstract
Introduction	12	8
1 Current status in the Internet of Things (IoT)	20	16
1.1. IoT - Key features	20	16
1.1.1. Taxonomy	20	16
1.1.2. Advantages of IoT	24	17
1.1.3. Trends in technological evolution	25	18
1.2. IoT Architecture	26	18
1.2.1. Complexity	27	19
1.2.2. Fog Computing	28	19
1.2.3. Frameworks	28	20
1.2.4. Technologies for IoT	29	20
1.2.5. Confidentiality and identity protection	30	20
1.2.6. Security aspects	31	20
1.3. Internet of Multimedia Objects (IoMT)	32	21
1.3.1. Concept	32	21
1.3.2. The role and main characteristics of IoMT	34	21
1.3.3. Field of applicability and standardization of IoMT	35	22
1.3.4. Challenges in developing IoMT	38	22
1.4. Chapter Summary	39	23
2 Services, protocols and connectivity in IoT	40	24
2.1. Service orientation in IoT	40	24
2.1.1. Service Oriented Architecture (SOA)	40	24
2.1.2. REST - Representational state transfer	43	25
2.1.3. Microservices	46	25
2.2. Event based architecture	48	26
2.2.1. Concept and architectural model	48	26
2.2.2. Event-oriented service architecture	51	26
2.3. Communication protocols for IoT	51	27
2.3.1. HTTP (Hypertext Transfer Protocol)	53	27
2.3.2. DDS (Data Distribution Service)	54	27
2.3.3. AMQP (Advanced Message Queueing Protocol)	55	28
2.3.4. CoAP (Constrained Application Protocol)	55	28
2.3.5. MQTT (Message Queue Telemetry Transport Protocol)	56	28
2.3.6. XMPP (Extensible Messaging and Presence Protocol)	57	28
2.4. Sensors connectivity to the Internet and Web	58	29
2.4.1. Sensor Integration	58	29
2.4.2. Modeling data from sensors	58	29
2.4.3. Sensor Networks	59	29
2.5. Contribute to real-time data analysis using web services and metadata	60	30
2.5.1. Metadata and dynamic semantic marking	60	30

2.5.2.	Web services and location of related processing	62	32
2.5.3.	Real-time data statistics and analysis	64	32
2.6.	Chapter Summary	66	34
3	IoT approach as a Social Network	67	35
3.1.	Social IoT	67	35
3.1.1.	Concept and system description	67	35
3.1.2.	How can social links improve IoT communications	68	36
3.1.3.	Relay discovery and selection of peer nodes	69	36
3.1.4.	Communication mode selection	71	37
3.1.5.	Resources management and allocation	73	38
3.2.	Web of Things (WoT)	74	38
3.2.1.	Concept	74	38
3.2.2.	Architecture	75	38
3.2.3.	Methods and technologies used	77	39
3.2.4.	Web of Things security threats	78	39
3.3.	Dynamic composition method of collaborative services in IoT	79	40
3.3.1.	SIoT demonstrator - IoT platform for integrating and testing collaborative services	79	40
3.3.2.	Architecture	80	40
3.3.3.	Application layer	82	41
3.4.	Chapter Summary	86	44
4	Integration of IoT into Cloud via web real-time communications (Web-RTC)	87	45
4.1.	Web real time communications	87	45
4.1.1.	WebRTC architecture	87	45
4.1.2.	WebRTC API	88	46
4.1.3.	STUN and TURN protocols	92	47
4.1.4.	Signaling in WebRTC	94	47
4.1.5.	Advantages to use WebRTC technology	94	48
4.1.6.	Libraries and WebRTC Toolkits	95	48
4.2.	Contribution to real-time peer-to-peer communication	96	49
4.2.1.	Development and call control environment	96	49
4.2.2.	Connection management	98	50
4.2.3.	Data flow management	99	51
4.2.4.	Graphical user interface	99	51
4.3.	Contribution to Cloud authentication by processing multimedia objects	101	52
4.3.1.	Technologies and implementation	101	52
4.3.2.	Neural networks workflow approach	101	52
4.3.3.	Cloud integrated services approach	104	53
4.4.	Chapter Summary	108	54
5	Web Services for Telematics in IoT	109	55
5.1.	Contributions to telemetry using virtual instrumentation and web services	109	55
5.1.1.	Data transmission from smart sensors and transducers	110	56
5.1.2.	The Cloud Instrumentation Server and TCP based telemetry	111	57
5.1.3.	Key features of the solution in the Big Data context	115	57

5.1.4.	Web based published control panel	116	58
5.1.5.	Implementation of web presentation services	117	58
5.2.	Contribution to responsive access to the terminal in IoT	120	59
5.2.1.	Concept	120	59
5.2.2.	Reactivity and responsiveness	121	60
5.2.3.	The "CON-INTEL Dashboard"	125	61
5.3.	Chapter Summary	130	62
6	Web development and semantic management for IoT	131	63
6.1.	Semantics and ontologies	131	63
6.1.1.	Semantic Web	131	63
6.1.2.	Semantic Web Solutions	132	63
6.1.3.	Challenges of the Semantic Web	132	63
6.2.	Standards and taxonomies	133	64
6.2.1.	Resource Description Framework	134	64
6.2.2.	RDFa (Resource Description Framework in Attributes)	135	64
6.2.3.	Microformat	138	65
6.2.4.	Microdata	139	65
6.2.5.	JSON-LD (JavaScript Object Notation for Linked Data)	140	65
6.3.	Semantic Web of Things	141	66
6.3.1.	Semantic sensor networks	142	66
6.3.2.	Semantic modeling and data sensor annotation	143	66
6.3.3.	W3C semantic sensor networks	143	67
6.4.	Contribution to the accessibility and management of semantic interfaces in IoT	145	67
6.4.1.	Accessible Rich Internet Applications	146	68
6.4.2.	AMP - Accelerated Mobile Pages	150	69
6.5.	Contribution to resource management using progressive and inter-platform offline capabilities	152	70
6.5.1.	Content to terminal adaptation	152	70
6.5.2.	Management through progressive web methodology	154	71
6.5.3.	Offline capabilities	156	71
6.6.	Chapter Summary	159	73
7	General conclusions and original contributions	161	75
7.1.	Main original contributions	162	76
7.2.	Future research directions	167	81
7.3.	Validation and dissemination of scientific results in publications and research projects	168	82
7.3.1.	Published works indexed ISI	168	82
7.3.2.	Published works to be indexed ISI	170	83
7.3.3.	Published works indexed in international databases (EBSCO)	170	83
7.3.4.	Monographs and chapters of scientific books	171	84
7.3.5.	Research projects	171	84
	Bibliography	171	85
	Short abstract (romanian/english)	xlvi	i
	CV (romanian/english)	xlvii	iii

Introducere

Internetul lucrurilor ("Internet of Things" - IoT) s-a impus inițial ca un concept generos de inter-conectare *nemijlocită* (uman), *cu ajutorul Internetului*, a unei multitudini de dispozitive, instrumente, echipamente – de exemplu electrocasnice, sisteme mobile vehiculare ("connected cars" etc) care comunică între ele dar și cu infrastructura fixă ș.a.m.d. [1]. În traducerea românească, prin extensie, "Internetul Obiectelor" poate include nu numai obiecte fizice (hardware) ci și obiecte software (inclusiv în sensul programării *orientate pe obiecte*) și, de asemenea, obiecte virtuale – mergând chiar până la "avatare" ale lucrurilor - modele comportamentale care sunt *reprezentante* (tranzacționale) ale obiectelor fizice etc.

Încorporate cu componente electronice, conectivitate la Internet și alte forme de hardware (cum ar fi senzori), aceste dispozitive pot comunica și interacționa cu alte dispozitive și obiecte prin Internet și pot fi monitorizate și controlate de la distanță.

Definirea Internetului lucrurilor/obiectelor a evoluat datorită convergenței mai multor tehnologii, analizelor în timp real, învățării inteligente („machine learning”), senzorilor și sistemelor integrate. În sfera industrială Internetul obiectelor se referă la senzori, instrumente și alte dispozitive interconectate cu aplicațiile industriale ale computerelor, incluzând (dar fără a se limita la) gestionarea energiei electrice și managementul producției [2]. Această conectivitate permite colectarea, schimbul și analiza datelor, vizând facilitarea îmbunătățirii productivității și eficienței, precum și alte beneficii economice.

Conceptul IoT s-a confruntat cu o critică proeminentă, în special în ceea ce privește problemele legate de confidențialitate și securitate legate de aceste dispozitive și de intenția lor de prezență pervazivă. Tendința majoră semnificativă a IoT în ultimii ani este creșterea explozivă a dispozitivelor conectate și controlate de Internet [3]. Gama largă de aplicații pentru tehnologia IoT înseamnă că specificul poate fi foarte diferit de la un dispozitiv la altul, dar există caracteristici de bază comune celor mai multe.

IoT creează oportunități pentru o integrare mai directă a lumii fizice în sisteme bazate pe computer, ceea ce duce la îmbunătățiri ale eficienței, beneficii economice și eforturi umane reduse. Numărul de dispozitive IoT a crescut cu 31% față de anul precedent la 8,4 miliarde în anul 2018 și se estimează că vor exista 30 de miliarde de dispozitive până în 2020. Valoarea globală a pieței Internetului va fi estimată la 7,1 trilioane de dolari până în 2020 [4].

O tendință distructivă în IoT este ca puncte finale de acces și gestiune a aplicațiilor să fie îmbinate împreună într-un întreg coeziv. În acest sens asigurându-se o soluție optimizată pentru un utilizator indiferent de dispozitivul pe care îl accesează aplicația, fie că este vorba despre un laptop, mașină sau chiar un frigider inteligent. Acesta înseamnă că un utilizator poate avea o experiență consistentă pe măsură ce comută între dispozitive. De exemplu, un utilizator ar putea accesa o rețetă online prin intermediul ecranului de pe frigiderul inteligent

și apoi transferând experiența de utilizare pe ecranul autoturismului în drum pentru a cumpăra ingredientele [5].

Inteligența ambientală și controlul autonom nu fac parte din conceptul original al Internetului obiectelor. Inteligența ambientală și controlul autonom nu necesită neapărat și structuri Internet. Cu toate acestea, există o schimbare în cercetare (de către companii precum Intel) pentru a integra conceptele IoT și controlul autonom, cu rezultate inițiale spre această direcție, considerând obiectele ca forță motrice pentru un Internet al obiectelor autonom. O abordare promițătoare în acest context este învățarea profundă („deep learning”) a consolidării în care majoritatea sistemelor IoT oferă un mediu dinamic și interactiv.

Inteligența IoT poate fi valorificată la trei niveluri: dispozitive IoT, noduri Edge/Fog și Cloud computing. Necesitatea unui control inteligent și a unei decizii la fiecare nivel depinde de sensibilitatea timpului aplicației IoT. De exemplu, camera autovehiculului autonom trebuie să facă detectarea obstacolelor în timp real pentru a evita un accident. Această luare a deciziilor rapide nu ar fi posibilă prin transferarea datelor de la vehicul la noduri Cloud și returnarea previziunilor înapoi la vehicul. În schimb, toată operațiunea ar trebui să fie efectuată local în vehicul. Integrarea algoritmilor de învățare automată avansată, inclusiv învățarea profundă în dispozitivele IoT, este o zonă activă de cercetare pentru a face obiectele inteligente mai aproape de realitate. În plus, este posibil să se obțină cea mai mare valoare din implementările IoT prin analizarea datelor IoT, extragerea informațiilor ascunse și precizarea deciziilor de control [6].

În viziunea Internetului obiectelor, obiectele creează un sistem omniprezent și interconectat care utilizează mai multe tehnologii de comunicații (de obicei pe termen scurt). Etichetele RFID au fost unul dintre primele exemple din acest domeniu. Cu toate acestea, în timp au apărut noi tehnologii care fac comunicarea între obiecte mai eficientă. Printre acestea se numără standardul IEEE 802.15.4 și, mai presus de toate, recenta sa modificare IEEE 802.15.4e, capabilă să crească considerabil fiabilitatea conexiunilor de radiofrecvență și eficiența energetică. Aceste tehnologii de nivel inferior, integrate în arhitecturi de protocol bazate pe protocolul IP, reprezintă “coloana vertebrală” a Internetului obiectelor, fiind capabile să comunice cu nodurile rețelei de Internet [7]. În acest sens, este important să menționăm protocoalele IETF 6LoWPAN, RPL și CoAP, capabile să creeze în mod operațional o rețea IP de obiecte care pot comunica cu Internetul pentru a crea noi servicii în mai multe domenii de aplicație.

Starea actuală a piețelor IoT (Internet of Things) și M2M (Machine to Machine) este extrem de fragmentată. IoT și M2M au soluții verticale concepute independent și separat pentru aplicații diferite.

Aceste soluții verticale și izolate au impact asupra implementărilor pe scară largă ale IoT, precum cele necesare pentru scenariile emergente, cum ar fi orașele inteligente. De exemplu, face practic imposibilă garanția de securitate, descoperirea serviciilor, gestionarea dispozitivelor etc. într-un cadru eterogen de miliarde de dispozitive.

Starea și evoluția actuală a IoT și M2M este și va fi orientată pe semantică. Aceste servicii trebuie să fie definite cu descrieri semantice a resurselor. Astfel, conținutul poate fi înțeles și reutilizat de aplicații și organizații diferite, pentru a ajunge la o colaborare între diferiții furnizori și utilizatori. Alta direcție evolutivă este integrarea recunoașterii emoțiilor și activarea serviciilor IoT corespunzătoare emoțiilor în obiecte Web. Sprijinul mai bun bazat pe ontologii și creații bazate pe cunoaștere semantică îmbunătățesc serviciile de conștientizare emoțională realizată prin combinarea mai multor microservicii. Această abordare a integrării emoțiilor armonizează serviciile IoT în fiecare aspect al activității umane [8].

SIoT (Social Internet of Things), SWoT (Social Web of Things) și standardele descrise (Open standards) reprezintă modul în care se poate ajunge direct la interoperabilitate și colaborare. Tendința evolutivă a IoT este de asemenea orientată spre Cloud Computing. Prin urmare, Cloud-ul va reprezenta un factor cheie pentru stocarea conținutului, punct de întâlnire între aplicații diferite pentru inter-schimbul de date și construirea serviciilor.

În contextul lucrării de față - al unui demers integrativ de management al complexității - așa cum semnalizările/notificările "urcă" spre HTTP (ca protocol unificator) și expunerea de capabilități "urcă" spre nivelul Prezentare al stivei OSI. Interfața cu utilizatorul *transcende* factorul uman prin potențarea etichetelor *markup* care devin și adnotări semantice dar și *attribute* de control bazat pe evenimente. Intermedierea semantică - "brokerajul" între dispozitive IoT atât de diverse ca protocoale și cozi de mesaje - "urcă", se contopește în nivelul Prezentare, iar viziunea mea și "deschiderea" acestei teze de doctorat este *utilizarea viitoare a limbajelor de marcare pentru a programa direct gestiunea IoT*, SIoT (Social IoT), WoS (Web of Sensors), WoT (Web of Things), SWoT (Semantic WoT) etc - o abordare cuprinzătoare (cum cuprinzătoare a fost și gestiunea conținutului, adaptarea la terminale, receptivitatea / reactivitatea serviciilor web).

Oportunitatea și actualitatea tezei de doctorat

Interoperabilitatea va crește în mod semnificativ, iar resursele IoT vor fi refolosite în afara cazurilor de utilizare și a domeniului în care acestea au fost inițial implementate și concepute. Mai mult, descriere semantică a datelor va avea un impact puternic în scenariile actuale, cum ar fi eHealth, Smart Cities și Smart Grid, în cazul în care aplicațiile ar putea avea nevoie să schimbe cantități uriașe de date fără să știe cum sunt reprezentate [9]. Viitorul este imprevizibil, dar puterea datelor furnizate de toate resursele care sunt conectate la Internet va aduce o nouă concepție asupra lumii, unde semanticul este necesar pentru a descrie cât mai detaliat un mediu cu un grad ridicat de eterogenitate. Informațiile accesibile dinamic și accesibile interactiv reprezintă o inovație în sociologia informației într-o măsură similară cu cea a Internetului în sine.

Datele masive produse de resursele IoT și M2M colectate prin intermediul comunicațiilor și al partajării serviciilor trebuie să fie exploatate eficient. Această exploatare a datelor este

următoarea etapă – după Web-ul Semantic al Lucrurilor (SWoT) – prin tehnologii precum "Big Data" - "Data Mines" ("masive de date") sau "Machine Learning" (învățare asistată – învățare automată) care vor oferi soluții prin analiza inteligentă a datelor/resurselor [10]. "Data Mining" și "Data Analytics" (analiza datelor) sunt direct dependente de furnizarea de informații semantice de către Web-ul Semantic al Lucrurilor.

Ceea ce este cel mai important pentru a înțelege amploarea IoT este *diversitatea de aplicații*, cu mare plus-valoare, de la o ameliorarea stării generale de sănătate la creșterea siguranței în trafic. Valoarea IoT nu stă doar în ceea ce sa făcut până în prezent, ci în marea deschidere către viitor. Dezvoltatorii au proiectat aplicații care comunică cu dispozitive, mașini și obiecte. Ceea ce s-a schimbat este *puterea resurselor distribuite* pe care aceștia o pot mobiliza pentru a procesa și a stoca date în IoT și raportul calitate/cost al dispozitivelor înseși. Pe măsură ce aceste tehnologii tot mai eficiente din punct de vedere al costurilor, un factor hotărâtor pentru IoT o să devină bazele de date (SQL, noSQL și hibride) stocate în Cloud, astfel încât inginerii de software care doresc să exploateze din plin potențialul acestor dispozitive - fie că este vorba despre un robot, o dronă sau de un simplu termostat - trebuie să se concentreze asupra manipulării datelor. Pentru a încadra și mai bine această lucrare, pentru a înțelege și mai bine actualitatea și interconstrucțiile temei, trebuie să înțelegem motivele pentru care, în ultimii ani, atenția specialiștilor ICT s-a deplasat *de la Centrele de Calcul la Centrele de Date*.

Construirea de modele complexe - de la concepte la interacțiuni – poate fi ea însăși o provocare dată fiind complexitatea domeniului IoT. De aceea, *modelele cognitive* reprezintă un important mijloc de îmbunătățire a performanțelor IoT, prin implementarea unor capacități de conștientizare ("awareness" – adaptare la context) și de învățare în dispozitive și în mediile de interacțiune. Acest salt poate face ca obiectele "să se înțeleagă", să interacționeze între ele și cu utilizatorii într-un limbaj natural – generalizarea "controlului pe bază de mesaje" (în locul / în completarea "controlului pe bază de registre" tradițional) este și ea o importantă oportunitate.

O tendință de integrare a IoT este răspândirea API ("Application Programming Interfaces") care devin "conectori logici" ce permit aplicațiilor să comunice cu dispozitivele IoT ale fiecărui producător. APIs expun capacități care permit dispozitivelor să transmită date către aplicații, acționând ca un *middleware* și, în celălalt sens, pot permite aplicației să controleze dispozitivul și să servească drept interfață funcțională.

Inginerii de software care construiesc aplicații IoT utilizând API au nevoie de o modalitate rapidă de a furniza resurse pentru a transmite cantitățile masive de date spre stocare, iar faptul că „norii de calcul” – Cloud publici – sunt deja orientați către API este un mare avantaj. În esență, specialiștii IoT aleg o abordare API sau orientată spre servicii pentru dezvoltare. Acest demers vizează crearea unor servicii "compozite": servicii Cloud / API și servicii dispozitiv / API. Prin adoptarea acestei abordări, se pot trata cererile de servire separat de complexitatea proceselor de stocare a datelor și de comunicații în Cloud.

Oportunitatea noilor tehnologii rezidă în combinarea dezvoltării în IoT cu instrumentele pentru gestiunea API și pentru managementul securității.

Obiectivele tezei de doctorat

Prezenta lucrare urmărește o re-așezare a serviciilor web în profunzimea stivei OSI, o extindere a managementului IP – de sus-în jos ("top-down") – dar și o expunere a controlabilității – de jos în sus ("bottom-up") – o "publicare" a resurselor pentru a fi agregate/angrenate în scenarii de utilizare a IoT.

Într-o sinergie cu tehnologiile Cloud care tind să abordeze "ca serviciu" toate aspectele calculabilității, obiectivele principale (corespondente și cu structurarea tezei) sunt:

- Implementarea celor mai moderne tehnologii web la *nivelele Prezentare și Aplicație* pentru a spori accesibilitatea (inclusiv prin terminale cu performanțe medii și inclusiv pentru cei cu deficiențe senzoriale) la capabilitățile IoT. Soluțiile preconizate au avut în vedere integrarea sporită a reprezentării tip intrare-stare-ieșire a IoT *în browsere (navigatoare web) obișnuite, și/sau* extinderea acestora cu utilitare ("plugins") dedicate, beneficiind de *receptivitate, adaptarea conținutului la terminal și afișare progresivă*.
- Valorificarea *tehnologiilor "de timp" real* potențate de progresul recent în viteza comunicațiilor IP de bandă largă, din categoria "Web RTC" ("Real Time Communications"), cu accent pe *obiectele multimedia* care pot fi abordate astfel unificat, în *Internetul obiectelor multimedia* (IoMT – "Internet of Multimedia Things").
- Un demers de *modelare a datelor din IoT* și integrare a *analizei datelor* ("Data Analytics") cu extinderea metodelor consacrate în masivele de date ("Data Mines" – "Big Data") către IoT, prin soluții bazate pe *semantic web* (ontologii, mediere semantică, gestiunea meta-datelor) către analiza datelor în timp real ("Real-Time Data Analytics").
- Contribuții la *integrarea IoT prin metode ale Rețelelor Sociale* – către soluții și demonstratoare SIoT ("Social IoT") la nivele TRL ("Technology Readiness Levels") cât mai ridicate.
- Expunerea *"controlului ca serviciu"* (de la nivelele Transfer și Sesiune către partea de sus a stivei OSI) *într-o serie de soluții telematice* moderne, prin abordarea comunicațiilor industriale în perspectiva unificată IoT, cu servicii REST (Representational State Transfer) bazate pe HTTP pentru *managementul evenimentelor* în configurații complexe de informatică instrumentală și tele-măsurare în Cloud.
- Abordarea *securității IoT*, cu accent pe soluții moderne de autentificare bazate pe inteligența artificială.
- Validarea rezultatelor științifice ale cercetării doctorale prin elaborarea și publicarea de *articole sumative* care să evedențieze și locul contribuțiilor originale ale doctorandului.

Cercetarea a avut ca scop principal "coborârea serviciilor web spre hardware" și în domeniul IoT, un hardware caracterizat de capabilități relativ limitate: din punct de vedere al complexității se poate extinde paralela client ↔ "Thin Client" (client subțire) la paralela echipament de rețea (noduri de rețea) ↔ dispozitive IoT, așadar lucrarea a avut în vedere extinderea capabilităților *peer-to-peer* pentru a realiza "federații" / "colonii" și un management al infrastructurii IoT cu *metode sociale*.

Conceptul "Thin Client" ("client subțire" – cu resurse limitate) se pretează la coborârea în stiva OSI a tehnicilor consacrate inițial la nivelul Prezentare, încă o treaptă sau două în jos, prin *middleware* având la bază invocarea de la distanță a unor metode (RPC – Remote Procedure Calls).

Această re-așezare a granițelor dintre nivelele OSI e facilitată de noi soluții alternative: de exemplu, în loc să fie trimise datele la procese – în vechea paradigmă client-server – se pot trimite acum și procesele la date ; invers, în loc să se facă procesarea datelor la locul achiziției, datele achiziționate se pot trimite acum la servere de instrumentație în Cloud.

Pentru a "aplatiza" în cât mai mare măsură nivelele de ierarhizare, cercetarea doctorală are în vedere o folosire extinsă a tehnologiilor RESTful în care

- semnalizările (privind "evenimentele" – modelate ca "cereri de întrerupere") asupra unor sau "tranziții" în "mașina algoritmică de stare" / "automatul cu stări finite" ale unui serviciu web pentru IoT sunt *mesaje unificate HTTP/Siar*
- "stările" (inclusiv "stările de calcul" – considerate, generic, ca "rutine de servire a întreruperilor") sunt *blocuri constructive independente* ("atomizate" pentru un minimum de interdependențe ierarhice – "imbricări"), așa-zisele SIB (Service Independent building Blocks), cu o structură internă consolidată (de unde avantajul de a se reduce complexitatea testelor de integrare).

"Coborârea serviciilor web în stiva OSI" adâncind accesul prin browser-e obișnuite urmărește să extindă și asupra IoT conceptul de *Management IP*. Sunt gestionate astfel, în mod unificat, și obiectele fizice și cele virtualizate - în paradigma "Internetul obiectelor software". Un obiect poate fi reprezentat de un "avatar" ("alias") folosindu-se o abordare *tranzacțională*. Tranzacțiile se pot efectua în spații virtuale – desfășurându-se între acești reprezentanți ai obiectelor și nu direct între respectivele obiecte ; doar *deciziile de control* se pot întoarce spre obiectele propriu-zise – de exemplu până la nivelul fizic (nivelul dispozitiv IoT etc).

Dintr-un alt punct de vedere, noile tipuri de servicii web dezvoltate în cadrul cercetării doctorale se pot considera, în perspectiva tradițională, ca soluții menite a gestiona *publicarea*, numai că, pe de altă parte, *expunerea de capabilități* ale entităților IoT este reconsiderată ca "*publicare a resurselor*" complementată de o "*publicare a disponibilității*" (începând de la protocoalele de rețea cu advertising și până la abordarea cu Rețele Sociale – menționată anterior) mergând până la o publicare / raportare *a stărilor* (cu minim de redundanță și maximum de completitudine – sub aspectul de "integritate a stărilor").

Nu în ultimul rând, prezenta teză de doctorat urmărește să pună într-o nouă lumină competiția dintre serviciile web (cu avantajul major de a nu necesita instalarea pe dispozitive) și paradigma aplicațională ("app"): în ultimii ani a apărut tendința (pe alocuri exagerată) de înlocuire a serviciilor web cu "apps" - aplicații instalate pe terminalele mobile pentru orice facilitate, cât de mică, cu epuizarea rapidă a spațiului de stocare internă și, nu în ultimul rând, cu blocarea unor resurse când aplicația rămâne deschisă, alături de multe altele, chiar pe terminale „multi-core”.

Organizarea tezei de doctorat

Teza cuprinde o Introducere, precedată de o Listă de figuri (în număr de 85) și o Listă de abrevieri. Urmează 7 Capitle - primele 6 corespund obiectivelor propuse iar ultimul fiind dedicat concluziilor generale și contribuțiilor originale, evidențiind și validarea rezultatelor științifice în publicații și proiecte de cercetare. Finalul lucrării cuprinde bibliografia (196 titluri) și două anexe - Anexa 1 cu Demonstratorul „Contor Cloud” CON-INTEL și Anexa 2 cu 3 Lucrări reprezentative ale autorului. Pe parcursul tezei, pe lângă abrevieri (explicitate în lista dedicată) am folosit ca atare (în limba engleză) și o serie de termeni tehnici consacrați pe care i-am pus în ghilimele și/sau în paranteze.

- Partea introductivă prezintă oportunitatea și motivarea tematicii alese, precum și obiectivele propuse pentru activitatea de cercetare doctorală.
- Capitolul 1 intitulat "Stadiul actual în Internetul Obiectelor (IoT)" realizează o trecere în revistă a tehnologiilor, soluțiilor și proiectelor IoT. Prezentarea are în vedere și modelele de afaceri, dat fiind că "praful inteligent" – „smart dust” IoT a părut ani de zile să "le scape printre degete" producătorilor, derutați de tendința de a oferi dispozitive... gratuite și de a monetiza preponderent serviciile. O atenție deosebită e acordată problemelor de interoperabilitate (de "management al diversității" IoT) și valorii de întrebuințare pentru utilizatorii finali. Este descrisă paradigma IoMT (Internetul obiectelor Multimedia), ca un subansamblu specializat al IoT iar în final sunt evidențiate aspecte particulare privind exigențele tehnice ale IoMT ca și soluțiile pentru problemele specifice de procesare – precum sistemele multi-agent într-un Cloud care "ia în considerare" (în mod adaptiv) cerințele multimedia („multimedia-aware Cloud").
- Capitolul 2, "Servicii, protocoale și conectivitate în IoT" abordează aprofundat orientarea pe servicii în IoT, arhitectura bazată pe managementul evenimentelor, protocoalele de comunicații, conectarea senzorilor la Internet, cât și pe prezentarea contribuțiilor aduse la analiza de date în timp real ("real-time data analytics") folosind servicii și metadata web.
- Capitolul 3 având ca titlu "Abordarea IoT ca Rețea Socială", evidențiază conceptele de Internet Social al Obiectelor și Web al Obiectelor, prezentând arhitectura, metodele și tehnologiile folosite cât și problemele de securitate. În final sunt prezentate contribuțiile la compoziția dinamică a serviciilor, concretizate în

proiectarea și dezvoltarea unui demonstrator (pentru această metodologie) pe platformă IoT.

- Capitolul 4, "Integrarea IoT în Cloud prin comunicații web de timp real (Web-RTC)", descrie conceptul de comunicații web de timp real (și avantajele pentru IoT), prezentând arhitectura, interfața de programare a aplicațiilor (API), protocoalele, semnalizarea, bibliotecile și seturi de instrumente folosite. În finalul capitolului sunt detaliate două contribuții la comunicarea peer-to-peer în timp real și la managementul (prin Web-RTC) al recunoașterii faciale în Cloud.
- Capitolul 5, numit "Servicii web pentru Telematică în IoT", cuprinde identificarea aspectelor telematice în IoT și a problematicii specifice precum și prezentarea soluțiilor bazate pe servicii Web – în principal contribuții la telemăsurarea folosind instrumentație virtuală și servicii web și, de asemenea, accesul receptiv la IoT.
- Capitolul 6, intitulat "Dezvoltare web și managementul semantic pentru IoT", descrie managementul semantic al Internetului obiectelor prin abordarea ontologică – taxonomie, standarde, modelarea și adnotarea semantică a datelor – și evidențiază principalele contribuții la dezvoltarea și managementul accesibilității interfețelor semantice în IoT, ca și la gestiunea resurselor utilizând capabilități offline progresive și inter-platformă.
- Ultimul capitol al tezei prezintă concluziile generale și contribuțiile originale, direcțiile viitoare de cercetare, precum și validarea și diseminarea rezultatelor științifice în publicații și proiecte de cercetare.
- Bibliografia conține 196 de lucrări, între care sunt prezente și cele publicate de autorul tezei.
- Anexa 1 prezintă contribuția autorului la proiectul "CON-INTEL" prezentat în Capitolul 5, denumită "CON-INTEL Dashboard – Demonstrator Contor Cloud", iar Anexa 2 conține 3 Lucrări reprezentative publicate de autor în jurnalul Sensors (Q1) și în volumele indexate ISI ale unor conferințe internaționale de specialitate.

1. Stadiul actual în Internetul Obiectelor (IoT)

IoT permite o nouă profunzime în *automatică*: analiza distribuită a informației pentru o mai bună integrare sistemică, sporind granularitatea controlului și a tehnicilor de reglaj (o altă abordare a acoperirii și a preciziei). IoT utilizează tehnologiile existente și emergente pentru detectare, crearea de rețele ad-hoc și auto-organizare în robotică [11]. IoT exploatează progresele recente în software, scăderea prețurilor hardware și abordările tehnologice moderne [12].

1.1. IoT - Caracteristici cheie

Cele mai importante caracteristici ale IoT includ utilizarea dispozitivelor de dimensiuni reduse (senzori și traductoare inteligente etc.), conectivitatea prin legături active (*active engagement*) și inteligența artificială.

1.1.1. Taxonomie

Internetul obiectelor "gravitează" în jurul următoarelor categorii: *energie, comunicare, atribut funcțional, interfața utilizator locală, resurse hardware și software și costuri* [13]. Dat fiind că IoT este un domeniu emergent și în continuă evoluție, această taxonomie este extensibilă pentru a lua în considerare noile cazuri:

A. Energie

Pentru multe obiecte, energia este critică. În unele cazuri, poate chiar determina durata de viață a unui obiect.

B. Comunicare

Deoarece unele obiecte furnizează mai multe interfețe de comunicații, este posibil ca un dispozitiv să ofere câte un tip (mai multe tipuri) de comunicații pentru fiecare dintre aceste interfețe.

C. Atribute funcționale

1) Interacțiuni:

- **Senzor** - Obiectul poate extrage diverse date din mediul său cum ar fi temperatura, expunerea la lumină sau doar își transmite poziția (de exemplu, eticheta RFID).
- **Dispozitiv de acționare/actuator** - Obiectul poate acționa în mediul său: se poate deplasa, produce căldură, emite lumină etc.
- **Senzor și actuator** - Obiectul este un hibrid al celor două categorii anterioare: poate să adune date și să acționeze în mediul său. Ca atare, acesta moștenește specificul senzorilor și poate avea memorie sau nu.

2) **Mobilitate**: unele obiecte au fost proiectate cu scopul de a se mișca/deplasa (de exemplu, un pedomer) iar altele nu (de exemplu, un termometru).

- 3) **Management:** în funcție de cazul de utilizare, obiectele pot fi gestionate sau nu. În cazul unui obiect ușor de gestionat, o altă întrebare ar putea fi dacă managementul se efectuează prin acces fizic sau la distanță prin una dintre interfețele de comunicații.

D. Interfața locală cu utilizatorul

- 1) **Obiect:** Uneori, obiectul are componente care sunt dedicate unei interacțiuni directe cu utilizatorul său pentru utilizarea funcțiilor sale de bază etc.
- 2) **Utilizare directă:** Dacă obiectul are o interfață activă sau activă/pasivă, uneori este posibil ca utilizatorul să utilizeze cel puțin funcționalitățile sale de bază.
- 1) **Utilizarea prin punctul local de recepție:** Dacă obiectul comunică cu restul arhitecturii printr-un punct de recepție local, este posibil ca utilizatorul să utilizeze cel puțin funcțiile de bază ale obiectului prin punctul de recepție local.

E. Resurse hardware și software

- 1) **Resurse hardware:** cantitatea de memorie RAM, memorie, procesor (CPU), tipul de sursă de energie pe care se va baza etc.
- 2) **Fiabilitate:** Obiectul poate oferi mai multe mijloace pentru a se asigura că informațiile sale sunt corecte (de exemplu, un senzor deschis/închis care are două moduri diferite de a ști dacă ceea ce monitorizează este deschis sau închis) este mai sigur decât un obiect care oferă doar unul.
- 3) **Actualizări de software:** Dacă este posibil sau nu ca software-ul obiectului să fie actualizat.

1.1.2. Avantajele IoT

IoT poate realiza integrarea fără întreruperi a diverselor dispozitive echipate cu capacități de detectare, identificare, procesare, comunicare, acționare și rețea. Bazându-se pe un spațiu inteligent cibernetic, acesta deschide noi oportunități de piață și de producție [14].

În timp ce conectivitatea și achiziția de date sunt imperative pentru IoT, acestea nu ar trebui să fie un scop în sine ci un mijloc de evoluție tehnologică. Dintre toate tehnologiile, mentenanța predictivă este aplicabilă atât activelor existente cât și sistemelor de management. Obiectivul sistemelor inteligente de întreținere este reducerea timpilor de nefuncționare și creșterea productivității. Analizele industriale de mari dimensiuni vor juca un rol esențial în întreținerea predictivă a activelor, deși aceasta nu este singura capacitate a datelor industriale mari.

Sistemele ciber-fizice (*CPS - cyber-physical systems*) reprezintă baza tehnologică pentru comunicațiile industriale de anvergură având rolul de interfață între lumea reală și lumea cibernetică (*cyber-space*) [16]. Sistemele ciber-fizice pot fi proiectate urmând arhitectura 5C (*conexiune, conversie, ciberspațiu, cunoaștere, configurare*) și vor transforma datele colectate într-o informație decizională și eventual vor interfera cu activele fizice pentru a optimiza procesele [17].

În figura 1 este prezentată evoluția IoT.

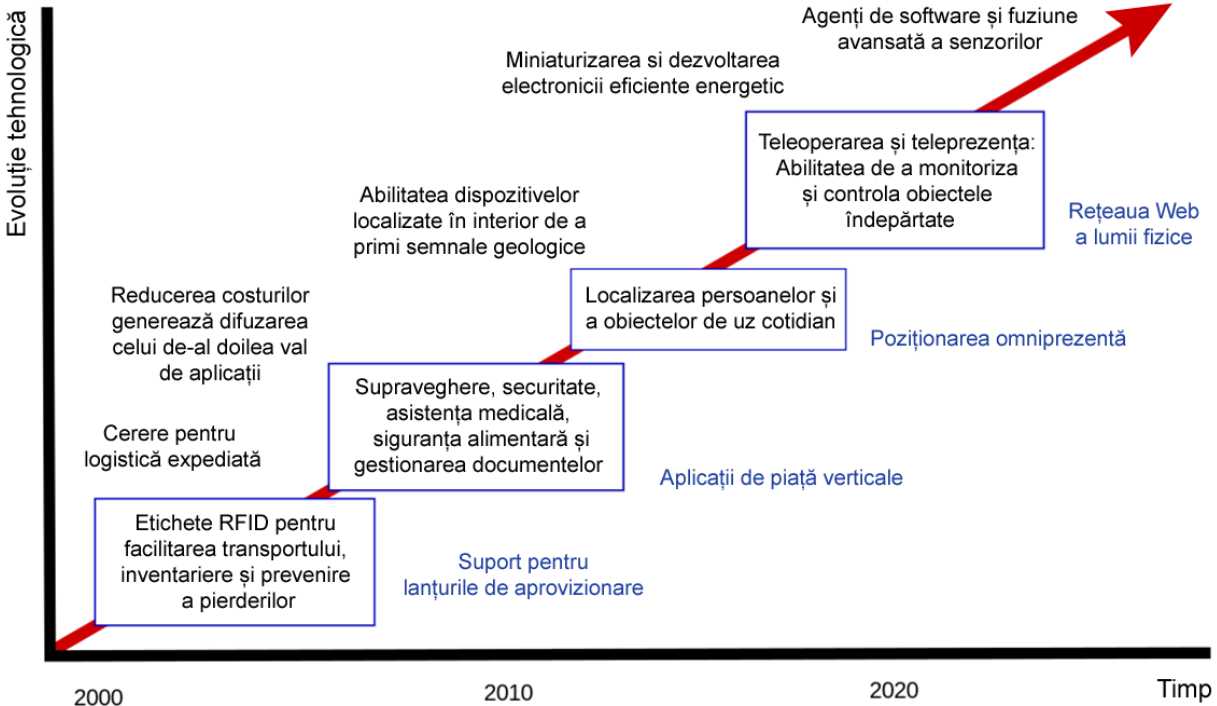


Figura 1: Grafic al evoluției IoT

1.1.3. Tendințe de evoluție tehnologică

Tendința majoră semnificativă a IoT în ultimii ani este creșterea explozivă a dispozitivelor conectate și controlate de Internet. Gama largă de aplicații pentru tehnologia IoT înseamnă că specificul funcțional poate fi foarte diferit de la un dispozitiv la altul, dar există caracteristici de bază comune.

IoT creează oportunități pentru o integrare mai directă a lumii fizice în sisteme informatice, ceea ce are ca rezultat îmbunătățirea eficienței, beneficii economice și eforturi umane mai reduse. Inteligența ambientală și controlul autonom nu au făcut parte inițial din conceptul de IoT. Cu toate acestea, există o schimbare de tendință în cercetare (de către companii precum Intel) pentru a integra conceptele de IoT și controlul autonom, cu rezultate inițiale spre această direcție, pentru a genera autonomie în IoT.

1.2. Arhitectura IoT

Arhitectura sistemelor IoT este alcătuită din trei niveluri:

- Nivelul 1: Dispozitive
- Nivelul 2: "Edge" Gateway
- Nivelul 3: Norul ("Cloud")

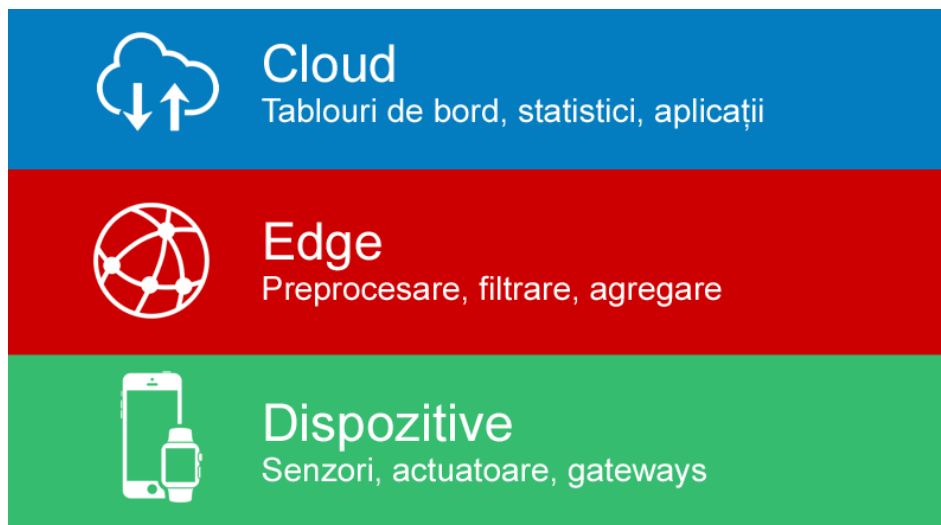


Figura 2: Grafic al arhitecturii IoT

1.2.1. Complexitate

IoT este adesea luat în considerare și studiat ca un sistem complex datorită numărului mare de legături diferite, interacțiunilor dintre actorii autonomi și capacității de integrare a noilor actori. Pe plan global complexitatea este probabil văzută ca un mediu haotic (comparativ cu sistemele caracterizate întotdeauna de finalitate). Ca o abordare practică, nu toate elementele din Internetul obiectelor se desfășoară într-un spațiu public global. Sub-rețelele sunt adesea utilizate pentru a diminua riscurile privind confidențialitatea, controlul și fiabilitatea. De exemplu, robotica domestică (domotică) care rulează într-o casă inteligentă poate să partajeze numai date și să fie disponibilă printr-o rețea locală. Gestionarea și controlul rețelei dinamice ad-hoc IoT este o sarcină dificilă în arhitectura rețelelor tradiționale – noline rețelele definite software (SDN - *Software Defined Networks*) oferă o soluție dinamică agilă care poate face față cerințelor speciale ale diversității aplicațiilor IoT inovatoare [20].

1.2.2. Fog Computing

"Fog Computing" este o alternativă viabilă pentru a preveni inundarea Internet-ului cu fluxuri de date. Puterea de calcul a dispozitivelor de tip Edge poate fi utilizată pentru analiza și prelucrarea datelor, oferind astfel o scalabilitate ușoară în timp real [23].

Fog Computing ("ceată computațională") – cunoscută și sub denumirea de Fog Networking, este o arhitectură care folosește dispozitive de tip Edge („*Entry Point*”) pentru a efectua o cantitate substanțială de calcul, stocare și comunicații la nivel local direcționate pe infrastructura Internetului. Fog Computing constă în fragmentarea calculului distribuită "cu granulația fină" a particulelor de ceață ("nori coborâți pe pământ", la extremitatea – *Edge* – a infrastructurii) în noduri Edge care efectuează în mod direct intrarea / ieșirea (adesea pentru a obține intrarea de la senzorului și ieșirea către un afișaj) sau controlul complet al unor procese locale în buclă închisă. Poate de asemenea să utilizeze „norișori” mai mici de tip

"Edge Clouds" numiți adesea „*Cloudlets*” – localizați mai aproape de Cloud-ul centralizat și care sunt localizați în Centre de Date foarte mari.

1.2.3. Cadre de lucru (Frameworks)

Cadrele de lucru IoT contribuie la susținerea interacțiunii dintre "obiecte" și vor permite structuri mai complexe, cum ar fi *repartizarea calculului* și dezvoltarea de *aplicații distribuite*. În prezent, unele cadre IoT se concentrează asupra soluțiilor de înregistrare în timp real a datelor, oferind o bază pentru a lucra cu multe obiecte și a le face să interacționeze rapid. Dezvoltările viitoare ar putea conduce la crearea unor medii specifice de dezvoltare a software-ului, pentru a crea software ce va interacționa cu hardware-ul folosit în Internetul obiectelor [26].

1.2.4. Tehnologii pentru IoT

Există multe tehnologii care permit dezvoltarea IoT. Crucială pentru domeniu este rețeaua utilizată pentru comunicații între dispozitivele IoT, un rol pe care îl pot asigura mai multe tehnologii cu sau fără fir:

- Adresabilitate: Conceptul de auto-identificare se baza inițial pe etichete RFID (*Radio-frequency identification*) și identificare unică prin codul de produs electronic, însă aceasta a evoluat în obiecte care au o adresă IP (*Internet Protocol*) sau un URI (*Uniform Resource Identifier*).
- Comunicații fără fir de rază medie ("Medium-range wireless")
- Comunicații fără fir la mare distanță ("Long-range wireless")
- Prin cablu: Ethernet, Multimedia over Coax Alliance (MoCA) sau Power-line communications (PLC).

1.2.5. Confidențialitate și protecția identității

Unul dintre motoarele cheie ale Internetului sunt datele. Succesul ideii de conectare a dispozitivelor pentru a le face mai eficiente depinde de accesul, stocarea și prelucrarea datelor. În acest scop, companiile care lucrează în domeniul IoT colectează date din mai multe surse și le stochează în rețeaua lor Cloud pentru procesare ulterioară. Acest lucru lasă ușa larg deschisă pentru pericolele de confidențialitate, securitate și vulnerabilitate. Celelalte aspecte se referă la alegerea consumatorului, la proprietatea asupra datelor și la modul în care este utilizat. Deși încă în fază incipientă, reglementările privind aceste aspecte legate de confidențialitate, securitate și proprietate asupra datelor continuă să se dezvolte și depind de țară și zona politico-geografică. Acestea se concentrează pe următorii 3 factori [28]:

- **Securitatea datelor**
- **Consimțământul** (asupra utilizării datelor)
- **Minimizarea datelor**

1.2.6. Aspecte de securitate

Fiecare dispozitiv conectat creează oportunități pentru atacatori. Aceste vulnerabilități sunt largi, chiar și pentru un singur dispozitiv mic. Riscurile prezentate includ transferul de date, accesul la dispozitiv, dispozitivele defectuoase și dispozitivele întotdeauna conectate

care pot fi observate de cei neautorizați – ca mod de operare – spre deosebire de dispozitivele cu conectare intermitentă (doar pe duratele strict necesare).

Principalele provocări în domeniul securității rămân restricțiile de securitate asociate cu presiunea pe ieftinirea dispozitivelor și numărul tot mai mare de dispozitive – de canale care creează mai multe oportunități de atac [29].

1.3. Internetul obiectelor multimedia (IoMT)

1.3.1. Concept

"Internetului obiectelor multimedia" (IoMT) este o paradigmă nouă în care obiecte multimedia inteligente și eterogene pot interacționa și coopera între ele și cu alte dispozitive conectate la Internet pentru a facilita serviciile și aplicațiile bazate pe multimedia disponibile la nivel global pentru utilizatori.

Serviciile și aplicațiile moderne și emergente din ce în ce mai accesibile prin intermediul Internetului au lărgit în mod rețeaua globală – în prezent, peste 9 miliarde de dispozitive de rețea sunt conectate la Internet, facilitând dezvoltarea și operarea de servicii pentru comunicare (e-mail, rețele sociale, chat, blog, forumuri etc.), activități de divertisment (jocuri, cărți etc), cumpărături, schimb de cunoștințe (educație, informații geografice, enciclopedii etc.) pentru mai mult de 2,5 miliarde de persoane din întreaga lume [30]. Într-un sistem bazat pe IoT, inteligența și capacitatea de declanșare a acțiunii sunt încorporate în dispozitive cu ajutorul senzorilor și respectiv al dispozitivelor de acționare. În mod similar, un Cloud permite capacitatea de a dezvolta, menține și executa diferite servicii prin furnizarea de resurse scalabile de calcul și de stocare. Astfel, utilizatorii au posibilitatea să monitorizeze și să controleze dispozitivele de oriunde și oricând. Cu toate acestea, serviciile Cloud oferite de diferite servere (cum ar fi ThingSpeak, Xively, ioBridge, Carriots, Axeda) se limitează la un sistem simplu de gestionare a dispozitivelor și la înregistrarea datelor senzorilor. Serverele Cloud existente au fost implementate pentru procesarea *de date scalare* de la IoT, care au în principal caracter asincron și sporadic [34].

1.3.2. Rolul și caracteristicile principale ale IoMT

Rolul și caracteristicile principale ale IoMT, se definesc prin două sisteme care sunt utilizate în prezent pentru a implementa servicii în aplicațiile multimedia majore IoT: Sistemele multimedia wireless (Wireless Multimedia Systems), care sunt utilizate pentru monitorizarea mediului și unde dispozitivele de detectare pot primi *feedback* pentru a controla procesul de achiziție; și rețelele wireless pentru senzori multimedia (Wireless Multimedia Sensor Networks - WMSN), în cazul în care dispozitivele multimedia nu au capacitate (sau au capacitatea limitată) de a primi feedback.

În rețelele multimedia convenționale, dispozitivele multimedia raportează numai informațiile multimedia achiziționate din vecinătatea lor. Acest conținut multimedia poate fi transmis către utilizator sau stocat în Cloud pentru procesare și recuperare la cerere. În general, aceste dispozitive multimedia sunt proiectate pentru a putea comunica cu alte dispozitive multimedia cu caracteristici similare (dispozitive omogene), respectiv stive de comunicare similare, resurse similare etc.

În schimb, IoMT prevede ca dispozitivele multimedia, cum ar fi camerele sau microfoanele, să poată fi accesate global printr-o adresă IP unică, la fel ca și computerele și alte dispozitive de rețea conectate la Internet.

1.3.3. Domeniul de aplicabilitate și standardizare a IoMT

Ca urmare a reducerii progresive a mărimii și costului dispozitivelor senzoriale multimedia, se așteaptă ca rețelele bazate pe IoMT să fie utilizate la scară largă, unde dispozitivele multimedia sunt capabile să creeze legături ad-hoc cu dispozitivele învecinate. Activarea organizării și auto-gestiunii autonome a acestor rețele extrem de dinamice pune sub semnul întrebării soluțiile disponibile în prezent, care au fost proiectate și dezvoltate pentru modele de rețea previzibile. Astfel, IoMT se poate defini ca o rețea globală de lucruri multimedia interconectate, care sunt identificabile și adresabile în mod unic pentru a obține date multimedia sensibile sau pentru a declanșa acțiuni, precum și pentru a avea capacitatea de a interacționa și comunica cu alte dispozitive și servicii multimedia, cu sau fără intervenție umană directă.

Cantitățile uriașe de date multimedia sunt supuse diferitelor proceduri de procesare înainte de transmisie la dispozitivul multimedia – transformare, cuantificare, estimare, codare a entropiei etc., astfel încât conținutul să poată fi comprimat pentru a reduce cerințele privind lățimea de bandă în timpul transmisiei. Aceste procese sunt computațional complexe și consumă o cantitate semnificativă de energie. Cu toate acestea, pot fi utilizate diferite soluții promițătoare pentru comunicarea multimedia eficientă, precum detectarea compresivă, codarea video distribuită, etc., pentru a facilita achiziția și prelucrarea multimedia pe dispozitivul multimedia din IoMT.

Având în vedere aceste seturi de cerințe, tehnicile de codare video sunt clasificate în trei clase:

1. Codare video convențională;
2. Codare video distribuită;
3. Achiziție cu rată comprimată statistic ("compressive sensing / compressive sampling").

1.3.4. Provocări în dezvoltarea IoMT

Transmisia conținutului multimedia de la senzor la serverul de aplicații, aflat de obicei în Cloud, impune cerințe stricte de gestionare a traficului esențiale în cazul streaming-ului în timp real, care are caracter continuu și poate sau nu poate fi tolerant la întârzieri (în funcție de cerințele specifice, de exemplu, în sistemele de management inteligent eBusiness sau în sistemele automate de securitate publică).

Integrarea Cloud cu CPS poate oferi un număr enorm de servicii. De exemplu, utilizatorii pot fi autorizați să acceseze în mod omniprezent datele de la senzori aflați la distanță iar seturi de reguli pot fi implementate pentru a controla din Cloud funcționarea automată a actoarelor ("senzor și acționare ca serviciu" - *Sensing and Actuation as a Service, SAaaS*), pentru a controla gestionarea identității și politicilor ca serviciu (*Identity and Policy Management as a Service - IPMAaaS*), permițând accesul la analizele de timp real pentru streamingul de supraveghere video (*Video Surveillance as a Service - VSaaS*) ș.a.m.d.

Scopul dorit este reducerea implicării umane în IoMT până la un minim viabil – inclusiv în detectarea obiectelor și a mișcărilor, recunoașterea facială, identificarea plăcuțelor de înmatriculare, conștientizarea situațională și acustică, analiza criminalistică, detecția furtului sau a accesului interzis etc. În concluzie, câteva noi utilizări posibile ale IoMT sunt: telemedicină, interacțiune socială inteligentă, gestionarea inteligentă a afacerilor (eBusiness), securitatea publică automată etc.

1.4. Sumarul capitolului

În ciuda potențialului IoT, liderii industriei și consumatorii se confruntă cu obstacole în calea acceptării mai largi a acestei tehnologii. Chiar dacă soluțiile IoT s-ar preta la adoptarea timpurie, ele nu au încă interoperabilitate și, mai ales, nu sunt complet "monetizate" – nu sunt reguli comerciale specifice, nu sunt "Business Cases", nu e protejată proprietatea intelectuală asupra "prafului inteligent" al dispozitivelor IoT a căror valoare le "scapă printre degete" producătorilor, după ce ani de zile s-a discutat că până și smartphones vor fi distribuite gratuit și că vor fi taxate doar serviciile (sau nivelul QoS/QoE *rezervat* – fie că a fost utilizat sau nu). Studiile arată că proiectele IoT sunt adesea propulsate de inițiative tehnologice mai degrabă decât de inovarea modelelor de afaceri. Am prezentat în acest capitol IoMT ca subansamblu specializat al IoT pentru integrarea și cooperarea dispozitivelor multimedia eterogene cu capabilități și resurse distincte de achiziție, calcul și comunicații. În comparație cu IoT, realizarea IoMT are provocări suplimentare și cerințe stricte de performanță. Un Cloud adaptat la („conștient de”) multimedia („multimedia-aware Cloud”), combinat cu sistemele multi-agent, poate ajuta la implementarea unor servicii și aplicații avansate.

2. Servicii, protocoale și conectivitate în IoT

Tehnologiile de comunicații orientate pe servicii (SOC - Service-oriented communications) sunt proiectate pentru a fi utilizate cu ușurință în contextul arhitecturilor orientate spre servicii. Aceste tehnologii sunt, în general, bazate pe software și sunt construite mai mult ca o aplicație de afaceri decât un sistem tradițional de comunicații de tip PBX (*private branch exchange*) [41]. Sistemele de comunicații orientate spre servicii permit serviciilor lor să participe la procesele de afaceri. Acestea își pun serviciile la dispoziția altor aplicații de afaceri din cadrul SOA (*Service-oriented architecture*) și permit reutilizarea serviciilor. Scopul comunicațiilor orientate spre servicii este să permită mediului de afaceri să construiască comunicații în procesele lor business, permițând o colaborare mai raționalizată între oamenii din cadrul afacerii. De obicei, aceasta presupune că anumite servicii sunt furnizate în contextul unui furnizor de servicii SOA. Acest lucru este adesea sub forma unei suite de servicii web, dar poate fi, de asemenea, atașat la alte mijloace de partajare a serviciilor, cum ar fi un *enterprise system bus* (ESB).

Cercetările în domeniu au căutat să exploreze modalitățile în care tehnologia informației poate fi utilizată mai bine pentru a îmbunătăți eficiența și productivitatea în mediul business [42]; în practică, s-a căutat adoptarea de concepte de arhitecturi orientate spre servicii și restructurarea modului în care diferitele sisteme de afaceri utilizate în cadrul întreprinderii pot fi proiectate pentru a permite fiecărui sistem să profite de avantajele oricărui alt sistem. Maturitatea tehnologiilor de servicii web oferă un cadru real pentru a permite unui sistem să profite de serviciile unui alt serviciu în conformitate cu principiile unei arhitecturi orientate spre servicii. Sistemele complexe de afaceri pot lucra împreună pentru a oferi soluții în moduri foarte personalizate utilizatorilor finali. Din ce în ce mai mult, aplicațiile se adaptează nevoilor specifice ale utilizatorilor, în loc să fie nevoie ca utilizatorii să se adapteze la funcționalitatea disponibilă a aplicațiilor.

2.1. Orientarea pe servicii în IoT

2.1.1. Arhitectura orientată pe Servicii (SOA)

Pe măsură ce Internetul a crescut de la un forum pentru schimbul de informații într-o piață business, o tehnologie maturată permitea computerelor să tranzacționeze mai ușor. Din aceste rădăcini ale Internetului sa născut tehnologia de servicii web.

Dacă sistemele sunt decuplate în timp, este dificil de a asigura și integritatea tranzacțiilor; sunt necesare protocoale de coordonare suplimentare. Replicarea datelor în cadrul diferitelor sisteme oferă o cuplare slabă (în disponibilitate), dar creează probleme în menținerea coerenței (sincronizarea datelor).

Serviciile Web W3C pot utiliza protocolul SOAP prin HTTP, permițând interacțiuni mai puțin costisitoare (mai eficiente) pe Internet decât prin soluții brevetate cum ar fi EDI (Electronic Data Interchange) / B2B. Pe lângă SOAP prin HTTP, serviciile web pot fi implementate și pe alte mecanisme de transport fiabile precum FTP.

Termenul "serviciu web" descrie un mod standardizat de integrare a aplicațiilor bazate pe web utilizând standardele deschise XML, SOAP, WSDL și UDDI prin protocoale Internet. XML este formatul de date utilizat pentru a conține datele și a furniza metadatele în jurul acestora, SOAP este utilizat pentru a transfera datele, WSDL este folosit pentru a descrie serviciile disponibile și UDDI listează ce servicii sunt disponibile.

2.1.2. REST - Representational state transfer

Representational State Transfer (REST) – corespunde, în linii mari, cu ideea de a *reprezenta* HTTP *tranzițiile* (semnalizările, evenimentele, "întreruperile") într-un automat cu *stări* finite ("mașină algoritmică de stare" - ASM) – generic, acestea pot fi "stări de calcul" ("rutine de servire a întreruperilor" etc). Această idee îndrăzneată [49], [53], [131], [172] urcă implementarea ASM spre straturile OSI superioare, cu avantajul (dar și provocarea de securitate) de a nu mai folosi porturi proprietare sau mai puțin răspândite (care trebuie deschise în toate sub-rețelele IoT implicate) ci numai portul 80, HTTP.

Terminologie specifică:

- *Resursă*: este un obiect sau o reprezentare a unui obiect semnificativă pentru aplicație. Exemple de entități ar putea fi: o carte, un ordin, un mesaj și orice altă entitate care poate fi extrasă dintr-un anumit context. Conceptul de resursă este, prin urmare, foarte asemănător cu cel al obiectului în *programarea orientată pe obiecte*;
- *Colecții*: sunt colecții de resurse. De exemplu, *Smartphone* poate fi denumirea unei colecții de resurse de *dispozitive mobile*;
- *URL* (Uniform Resource Locator): este calea prin care poate fi obținută o resursă și unde pot fi efectuate acțiuni.

2.1.3. Microservicii

Arhitectura orientată pe microservicii a apărut ca o soluție dominantă pentru dezvoltarea de aplicații moderne care necesită dezvoltare și scalare rapidă. Ideea din spatele orientării pe microservicii, este o evoluție a SOA (Service Oriented Architecture), ca o arhitectura orientată spre servicii în mediul distribuit [51].

Arhitectura orientată pe microservicii este caracterizată prin structura aplicațiilor care sunt alcătuite dintr-un număr de servicii independente, fiecare concentrându-se pe un anumit aspect lucrativ ("mici servicii" lucrative, de unde și numele de "microservicii"), care comunică între ele pentru a realiza aplicații mai complexe.

Arhitectură:

Microserviciile sunt o extinderea a stilului arhitectural orientat pe servicii (SOA) – a se vedea și paragrafele 2.2.2 și 3.2.3. Sub aspectul resurselor mobilizate (cât mai puține) și al încărcării computaționale (cât mai reduse), într-o arhitectură de microservicii, serviciile trebuie să fie „thin” (subțiri) iar protocoalele ar trebui să fie "deschise" (populare) și „ușoare” (lightweight) – cu minimum de interconștinări și "imbricări" ierarhice [52]. Beneficiul descompunerii unei aplicații în diferite servicii mai mici este acela că îmbunătățește modularitatea și facilitează înțelegerea, dezvoltarea și testarea aplicației – într-o perspectivă durabilă, sustenabilă.

Comunicarea între microservicii se bazează pe protocolul HTTP prin API-ul RESTful, folosind date în format JSON sau XML. Microserviciile individuale sunt în general tratate în mod asincron, declanșate de un eveniment cum ar fi un apel API sau o intrare în „coada de așteptare”.

În imaginea următoare este prezentată evoluția de la era „monolitică” până la microservicii.

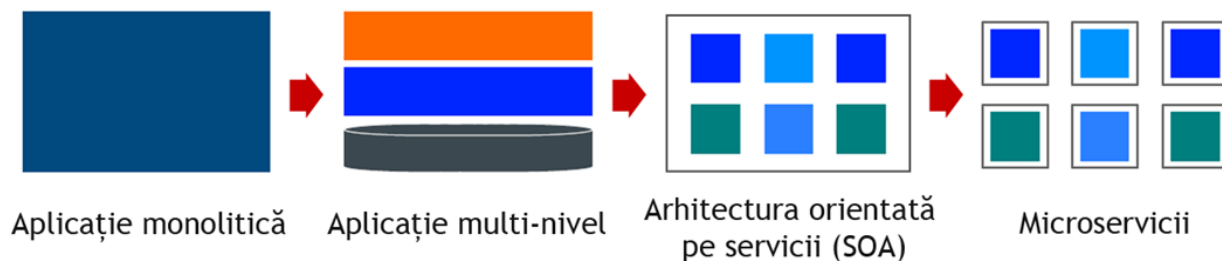


Figura 3: Evoluția arhitecturilor: "Monolitică", "Multi-nivel", "SOA", "Microservicii"

2.2. Arhitectura bazată pe evenimente

2.2.1. Concept și model arhitectural

Arhitectura bazată pe evenimente (EDA - Event-Driven Architecture) este un model de arhitectură software care promovează producția și consumul serviciilor prin orientare pe detectarea și reacția la evenimente [54] (o extensie a "controlului pe bază de întreruperi" tradițional).

Un eveniment poate fi definit ca "o schimbare semnificativă a stării". De exemplu, atunci când un consumator cumpără o mașină, starea mașinii se schimbă de la "*de vânzare*" la "*vândut*". O arhitectură de sistem a dealerului auto poate trata această schimbare de stare ca un eveniment a cărui apariție poate fi făcută cunoscută altor aplicații din cadrul arhitecturii. Din perspectiva formală, ceea ce este produs, publicat, propagat, detectat sau consumat este un mesaj (tipic asincron) numit *notificare asupra evenimentului* și nu evenimentul în sine, care este schimbarea de stare ce a declanșat emisia mesajului. Evenimentele nu "călătoresc", doar "apar" (și sunt semnalizate). O arhitectură bazată pe evenimente poate fi construită pe patru nivele logice, începând de la sesizarea unui eveniment, continuând cu crearea reprezentării sale tehnice sub forma unei structuri de eveniment și terminând cu o reacție sau un set de reacții la acest eveniment.

2.2.2. Arhitectura orientată pe servicii condusă de evenimente

Arhitectura orientată pe servicii condusă de evenimente ("Event-driven SOA" sau "SOA 2.0") - este o formă de arhitectură orientată pe servicii (SOA), care combină inteligența și proactivitatea arhitecturii bazate pe evenimente cu capacitățile organizaționale găsite în sistemele de servicii.

Programarea bazată pe evenimente SOA 2.0 este structurată în jurul conceptului de relații decuplate dintre producătorii de evenimente și consumatorii evenimentului: pentru un consumator de evenimente nu contează unde și de ce a avut loc evenimentul ci doar faptul că va fi invocat atunci când evenimentul a avut loc. Sistemele și aplicațiile care separă producătorii de evenimente de consumatorii evenimentului se bazează în mod obișnuit pe un dispecer de evenimente – un "canal" (o coadă de evenimente care acționează ca intermediar între producătorii de evenimente și agenții de procesare a evenimentelor).

2.3. Protocoale de comunicații pentru IoT

Inovațiile continue în domeniul hardware, software și soluțiile de conectare din ultimul deceniu au dus la extinderea Internetului obiectelor cu un număr de dispozitive conectate ce sporește spectaculos pe zi ce trece. Cantitatea imensă de informație generată de aceste dispozitive necesită găsirea unei arhitecturi adecvate a sistemului capabilă să proceseze și să stocheze toate datele. În timp ce arhitecturile bazate pe Cloud sunt utilizate în acest scop, noua paradigmă de Fog Computing (a se vedea paragraful 1.2.2) este concepută pentru a scala și a optimiza infrastructurile IoT. Studii și exemple de soluții de IoT bazate pe Cloud au arătat că Cloud computing are potențialul de a satisface multe cerințe IoT, cum ar fi monitorizarea serviciilor, procesarea complexă a fluxurilor de date senzoriale și sarcinile de vizualizare (- de prezentare). Pe de altă parte, soluțiile bazate pe Fog Computing sunt considerate potrivite pentru a aborda procesarea în timp real, răspunsul rapid la cereri de date fără problemele de latență, extinzând astfel capacitățile Cloud mai aproape de marginea rețelei [58] (o descentralizare, o "externalizare" a acestor capacități).

2.3.1. HTTP (Hypertext Transfer Protocol)

Hyper Text Transport Protocol (HTTP) este protocolul de bază client-server folosit pentru Web și cel mai compatibil cu infrastructura de rețea existentă, fiind utilizat zi de zi de către dezvoltatorii web. În prezent, versiunea cea mai populară a acestui protocol este HTTP/1.1. Comunicarea dintre un client și un server are loc printr-un mesaj de cerere/răspuns, clientul trimițând un mesaj de solicitare HTTP, iar serverul returnând un mesaj de răspuns, care conține resursa solicitată, dacă cererea a fost acceptată. În ultima perioadă, HTTP a fost asociat cu REST (a se vedea paragraful 2.1.2), pentru dezvoltarea serviciilor web bazate pe un stil arhitectural specific, spre a defini interacțiunea dintre diferitele componente. Combinația dintre protocolul HTTP și REST este lăudabilă, deoarece dispozitivele se pot conforma cu ușurință, datorită modului standard de creare, citire, actualizare și ștergere a datelor (așa-numitele operațiuni CRUD – „*Create, read, update and delete*”). Conform acestei mapări, operațiunile de creare, actualizare, citire și ștergere a resurselor corespund metodelor HTTP POST, GET, PUT și DELETE.

2.3.2. DDS (Data Distribution Service)

DDS este un standard de interoperabilitate bazat pe date în timp real, definit de Object Management Group (OMG), care utilizează un model de interacțiune publicare-abonare. Spre deosebire de alte protocoale de publicare-abonare, DDS este descentralizat și bazat pe comunicații peer-to-peer și, ca atare, nu depinde de componenta broker. În DDS, editorii și abonații pot comunica ca și colegi ("peers") prin intermediul magistralei de date, permițând schimbul de date asincron pe baza solicitărilor de moment ale acestora.

Entitățile principale din arhitectura DDS includ: *Domeniu* („Domain”), *Participantul la domeniu* („Domain Participant”), *Subiectul* („Topic”), *Editorul* („Publisher”), *Abonatul* („Subscriber”), *Scriitorul de date* („Data Writer”) și *Cititorul de date* („Data Reader”). Editorii și abonații sunt împărțiți în Domenii – entități virtuale conceptuale care permit izolarea comunicației în grupul nodurilor care au interese comune.

2.3.3. AMQP (Advanced Message Queueing Protocol)

AMQP este un protocol "open standard" (aflat încă la nivel de recomandare), definit pe paradigma *publicare-abonare*, conceput pentru a permite interoperabilitatea între o gamă largă de aplicații și sisteme diferite, indiferent de designul lor intern. Inițial a fost dezvoltat pentru mesageria business cu ideea de a oferi o soluție non-proprietară care poate gestiona o cantitate mare de schimburi de mesaje ce se pot executa într-un timp scurt într-un sistem. Această caracteristică de interoperabilitate a AMQP este semnificativă, deoarece permite ca diferite platforme, implementate în diferite limbaje de programare, să poată schimba mesaje, care pot fi utile în special în sistemele eterogene.

2.3.4. CoAP (Constrained Application Protocol)

Acest protocol a fost conceput de grupul de lucru IETF (Internet Engineering Task Force) pentru utilizarea în dispozitive "constrânse" de o capacitate limitată de procesare. Similar cu HTTP, una dintre caracteristicile sale cele mai importante este folosirea arhitecturii REST. Cu această caracteristică, CoAP acceptă paradigma cerere/răspuns la fel ca REST HTTP. CoAP este considerat un protocol mai lejer ("light") ca cerințe, astfel încât antetele, metodele și stările sunt codate binar, reducând astfel costul de protocol în comparație cu multe alte protocoale. Trebuie remarcat faptul că, din cauza lipsei de fiabilitate cauzată de utilizarea UDP care s-a dovedit a fi o problemă pentru unele medii, IETF a creat un document suplimentar care adaugă posibilitatea CoAP de a rula peste TCP [63].

2.3.5. MQTT (Message Queue Telemetry Transport Protocol)

MQTT este unul dintre protocoalele de mesagerie considerate mai "lejere" („lightweight”), bazat pe paradigma *publicare/abonare*, ceea ce îl face potrivit pentru dispozitivele cu resurse limitate și pentru condițiile de conectivitate care nu sunt ideale, cum ar fi lățimea de bandă mică și latența ridicată. MQTT a fost lansat de IBM, cu cea mai recentă versiune MQTT v3.1 adoptată pentru IoT de către OASIS. Datorită simplității sale și a unui antet de mesaje foarte mic comparat cu alte protocoale de mesagerie, este adesea recomandat ca soluție de comunicare în IoT.

Autentificarea este implementată de mulți brokeri MQTT, printr-unul din pachetele de mesaje MQTT de tip control, numit CONNECT. Brokerii solicită clienților ca atunci când trimit mesajul CONNECT, să definească combinația de *nume de utilizator/parolă* înainte de validarea conexiunii sau să refuze validarea în cazul în care autentificarea nu a reușit. În general, securitatea continuă să fie o preocupare pentru MQTT.

2.3.6. XMPP (Extensible Messaging and Presence Protocol)

XMPP este un protocol deschis de mesagerie standard, formalizat de IETF, și a fost inițial conceput pentru mesageria instant și schimbul de mesaje între aplicații. Este un protocol bazat pe text, construit pe limbajul *Extensible Markup Language (XML)*, care implementează atât modelul client-server, cât și modelul *publicare-abonare*, care rulează peste TCP [65]. În soluțiile IoT este proiectat pentru a permite utilizatorilor să trimită mesaje în timp real, pe lângă gestionarea prezenței utilizatorului.

XMPP permite aplicațiilor de mesagerie instantanee să obțină toate caracteristicile de bază, inclusiv autentificarea, criptarea "cap-la-cap" ("end-to-end") și compatibilitatea cu alte protocoale.

Deoarece XMPP a fost conceput inițial pentru mesageria instantanee, există unele deficiențe care ar trebui avute în vedere în cadrul IoT-Fog-Coud. Folosind XML, mărimea mesajelor îl face incomod în rețelele cu probleme de lățime de bandă. Un alt dezavantaj este lipsa de QoS.

2.4. Conectarea senzorilor la Internet și Web

2.4.1. Integrarea senzorilor

Integrarea în Internet a senzorilor cu constrângeri de resurse este dificilă, întrucât protocoalele de Internet, cum ar fi HTTP, TCP sau chiar IP, sunt implementate în mod ubicuu și sunt prea complexe și solicitante de resurse. Pentru a realiza integrarea, sunt necesare alternative ușoare care pot fi ușor convertite de la/la protocoalele Internet. CoAP - elaborat de grupul de lucru CoRE al IETF, care se ocupă de mediile de stocare cu impact redus - oferă o alternativă ușoară la HTTP folosind o reprezentare binară și un subset de metode HTTP (GET, PUT, POST și DELETE). În plus, CoAP oferă o fiabilitate a transportului prin recunoaștere și retransmisie. Pentru integrare fără probleme, proxy-urile inversate pot converti 6LoWPAN la IPv6 și CoAP/UDP la HTTP/TCP astfel încât datele senzorilor să poată fi accesate folosind aceste protocoale omniprezente [66]. De asemenea, clienții Internet ar putea utiliza în mod direct CoAP pe lângă UDP.

2.4.2. Modelarea datelor de la senzori

Integrarea senzorilor în Internet utilizând CoAP/HTTP permite deja multe aplicații în care dezvoltatorii interoghează și procesează datele furnizate de un set cunoscut de senzori. Cu toate acestea, o astfel de integrare manuală are limitele ei. Se cere o descriere a senzorilor și a datelor pe care le produc. Tehnica predominantă pentru reprezentările de cunoștințe de pe web, care pot fi citite de mașină, este RDF (Resource Description Framework – a se vedea paragraful 6.2.1), care reprezintă cunoștințele ca triplete (*subiect-predicat-obiect*). Un set de „triplețe” formează un graf în care subiectele și obiectele sunt „noduri” și predicatul este „laturi”. Din graful format de aceste triplete, se pot deduce proprietățile tranzitive [67]. Astfel de proprietăți sunt exprimate adesea folosind limbajul OWL (Web Ontology Language), unul dintre limbajele principale (folosind schema RDF) pentru a defini ontologii pe web (a se vedea capitolul 6):

2.4.3. Rețele de senzori

Conceptul de "sensor web" este un tip de rețea de senzori care este foarte bine adaptat pentru monitorizarea mediului. Sintagma "sensor web" este, de asemenea, asociată cu un sistem de detectare care utilizează puternic World Wide Web.

Termenul "sensor web" a fost folosit pentru prima oară de Kevin Delin de la NASA în 1997, pentru a descrie o nouă arhitectură a rețelelor de senzori fără fir, în care dispozitivele individuale ar putea acționa și coordonat, într-un ansamblu [68]. În acest sens, termenul descrie un tip specific de rețea de senzori: o rețea amorfă de platforme senzoriale distribuite spațial care să comunice fără fir între ele.

Această arhitectură amorfă este unică, deoarece este atât sincronă, cât și independentă de routere, făcând-o diferită de cele mai tipice scheme de rețea tip TCP/IP. O platformă pentru un senzor poate fi orbitală sau terestră, fixă sau mobilă și poate avea chiar și accesibilitate în timp real prin Internet. Comunicarea "nod-la-nod" este atât omnidirecțională, cât și bidirecțională, unde fiecare nod trimite date colectate către fiecare alt nod din rețea.

2.5. Contribuție la analiza de date în timp real folosind servicii și metadata web

În continuare va fi prezentat și propus un concept și o soluție specifică pentru analiza datelor în timp real ("real time data analytics") în vederea determinării caracteristicilor navigării și preferințelor utilizatorilor asupra conținutului, pentru a colecta date spre a îmbunătăți experiența Web. Soluția propusă are la bază împărțirea informațiilor în părți mici, slab conectate, pentru a reduce complexitatea aplicațiilor și efortul de integrare a acestora prin încapsularea lor într-un set de servicii reutilizabile, folosind metadata standardizate și servicii Web, pentru a genera statistici în timp real și pentru procesarea acestora într-un mod "inteligent".

2.5.1. Metadata și marcare semantică dinamică

Web-ul Semantic (a se vedea capitolul 6) este un pas în evoluția arhitecturii Web, care permite utilizatorilor să publice informații mai bogate în pagini web obișnuite, fără a se baza pe servicii centralizate. Standardizarea metadatelor web semantice este într-un proces continuu de dezvoltare, unele extensii sunt specificate complet, altele sunt în stadiu intermediar. În soluția propusă, specificațiile de metadata utilizate sunt Microformats și RDFa (așa cum este prezentat în figura de mai jos), ea fiind concepută în primul rând pentru oameni și în al doilea rând pentru „mașini” (computere), utilizând un set de formate de date simple bazate pe standarde existente și general acceptate, încapsulând elemente semantice în cadrul marcajelor HTML pentru a facilita dezvoltarea descentralizată, reprezentând convenții pentru numele elementelor HTML, attribute și valori ale atributelor, cu semantică bine stabilită.

```
<div class="vcard" xmlns:foaf="http://xmlns.com/foaf/0.1/"
about="#me">

I am <span class="fn" property="foaf:name">George Stelea</span>
from <span class="org" property="foaf:organization">
Transilvania University</span>
of <address class="adr">
<span class="locality">Brasov</span>,
<span class="country-name">Romania</span>.
</address><br>
Website: <a class="url" rel="foaf:homepage"
href="http://www.unitbv.ro/">www.unitbv.ro</a><br>
E-mail: <a href="mailto:george.stelea@unitbv.ro"
class="email v:email" rel="foaf:mbox">george.stelea@unitbv.ro
</a><br>
Phone: <a class="tel" property="foaf:phone"
href="tel:0040723123xxx">+40 723.123.xxx</a><br>
</div>
```

Figura 4: Metadata RDFa și Microformats într-o pagina web HTML

Se poate observa flexibilitatea acestui tip de marcare – astfel marcările Microformats și RDFa sunt independente unele de altele și sunt încapsulate într-un element "div" HTML pentru a descrie entitatea unui element. Marcarea este ușor de procesat de un program care cunoaște aceste convenții, iar datele extrase pot fi utilizate într-o varietate de scopuri de către un instrument care înțelege semantica datelor. Marcajul a fost formatat cu metode CSS (Cascaded Style Sheets), prezentate în figura următoare, fără a fi influențate de atributele și valorile atributelor care specifică metadatele, deoarece ele numesc doar convenții însoțite de valori specifice contextului particular.

```
<style type="text/css">
  div.vcard { border:4px double #ddd; border-radius:15px;
padding:10px 10px 28px 10px; width:400px; background:#f5f5f5;}
  img.photo {width:109px; float:left; margin-right: 8px;}
  span.fn {padding-top:9px; display:inline-block;}
  span.org {padding-top:9px;}
  address.adr {display:inline-block;}
  span.locality {font-style:italic;}
  span.country-name {font-style:italic;}
  a.url {padding-top:10px; display:inline-block; color:#10926e;}
  a.email {display:inline-block; color:#10926e;}
  a.tel {display:inline-block; color:#10926e;}
</style>
```

Figura 5: Reguli CSS utilizate pentru formarea marcajului HTML

Figura de mai jos prezintă vizualizarea frontală ("front-end") a elementului de tip bloc HTML, care este afișat în browserul web atunci când un utilizator vizualizează și interacționează cu aplicația web.



Figura 6: Vizualizarea frontală (front-end) a elementului de tip bloc HTML folosind marcare semantică

2.5.2. Servicii web și localizarea procesărilor aferente

Internetul este văzut ca o platformă software în care utilizatorii își controlează propriile date, făcându-l utilizabil la dispoziția altora prin instrumente de colaborare și servicii specializate care pot fi utilizate în mod transparent de către oricine. Am dezvoltat un parser folosind script-uri PHP și JavaScript pentru a stoca incremental într-o bază de date metadatele Microformats și RDFa accesate de utilizatori pe paginile web ale aplicațiilor. Pentru a sprijini integritatea tranzacțiilor și a datelor am ales MySQL, un sistem de management al bazelor de date relaționale de tip open-source așa cum este prezentat în imaginea următoare (dar conform cu cerințele aplicației, o bază de date orientată pe documente pentru date semi-structurate (ex: NoSQL) poate fi de asemenea utilizată). Metadatele stocate treptat în baza de date sunt expuse ca serviciu Web RESTful, unde rezultatul unei procesări generează returnarea reprezentării unei resurse web.

Modelul de dezvoltare a aplicațiilor RESTful pentru serviciile de web a fost ales deoarece necesită o infrastructură minimă, implementând protocolul HTTP și formatul de fișier standard JSON (a se vedea paragraful 2.1.2), compatibil cu cele mai multe platforme moderne și limbaje de programare.

Operațiunile care pot fi aplicate resurselor sunt GET, POST, PUT și DELETE.

În imaginea următoare este afișată o reprezentare a resurselor JSON.

```
{
  "id": 10,
  "name": "George Stelea",
  "profileImage": "profile.png",
  "organization": "Transilvania University",
  "address": {
    "street": "Bulevardul Eroilor 29",
    "postalCode": "500036",
    "locality": "Brasov",
    "country": "Romania" },
  "website": "www.unitbv.ro",
  "email": "george.stelea@unitbv.ro",
  "phone": "0040723123xxx" }
```

Figura 7: Reprezentarea unei resurse utilizând formatul JSON

Componenta serviciilor web este implementată independent de aplicație, această abordare este preferată din motive de scalabilitate și flexibilitate, astfel încât modificările sau îmbunătățirile ulterioare nu vor perturba sau afecta sistemul.

2.5.3. Statistici și analize de date în timp real

Spre deosebire de analizele/statisticile tradiționale în care datele sunt stocate și apoi prelucrate, conceptul de statistici în timp real analizează datele "încă în mișcare", imediat ce intră în sistem. Pentru a obține o analiză în timp real în soluția propusă, am utilizat serviciul Microsoft Azure Stream Analytics, un motor Cloud care permite configurarea, gestionarea și procesarea evenimentelor și fluxurilor de date de la senzori, aplicații web, etc. în timp real. Utilizând Azure Stream Analytics, volume mari de date pot fi examinate folosind un proces structurat de luare a deciziilor pe o logică predefinită, cu rezultate imediate disponibile pentru îndeplinirea sarcinilor aplicației, generarea alertelor și a fluxurilor de lucru automate,

efectuarea simulărilor, modelarea și optimizările bazate pe un set complet de date și nu doar probe [70]. Mai jos este prezentată arhitectura Microsoft Azure Stream Analytics.

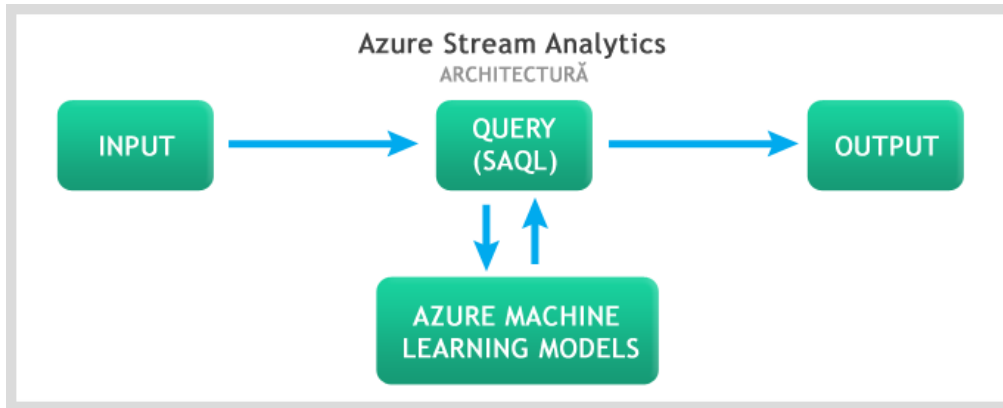


Figura 8: Arhitectura Microsoft Azure Stream Analytics

Rezultatele sunt afișate în "tabloul de bord" Microsoft Power BI, așa cum este prezentat în imaginea următoare.

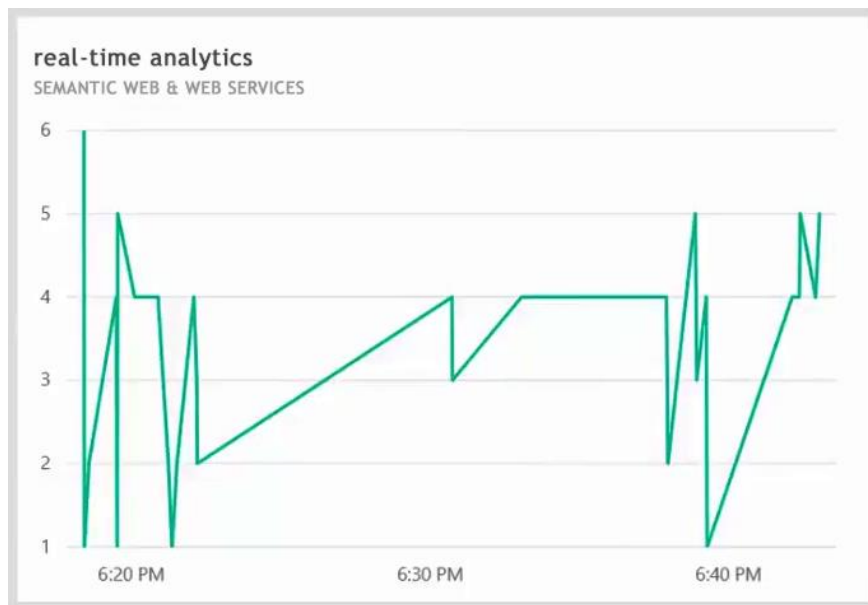


Figura 9: Tablou de bord pentru analize al Microsoft Power BI

Unul dintre obiectivele principale ale acestei soluții fiind simplitatea implementării sarcinilor și a portabilității, facilitând procesul de management și administrare, dezvoltarea a fost realizată în conformitate cu conceptul arhitecturii orientate pe microservicii.

Arhitectura orientată pe microservicii necesită adoptarea unui stil de dezvoltare a aplicațiilor, considerat ca un serviciu independent, care poate fi invocat de alte aplicații.

Microserviciile presupun crearea unor noi servicii pe baza celor existente, însă componentele sistemului ca un întreg trebuie să aibă un grad înalt de independență.

2.6. Sumarul capitolului

În acest capitol am realizat o analiză a serviciilor, protocoalelor (HTTP, DDS, AMQP, CoAP, MQTT, XMPP) și conectivității în IoT (cu precădere a fost detaliată conectarea senzorilor la Internet și Web, în perspectiva integrării, modelării datelor și a rețelelor). Conceptul de transformare, îmbinare și îmbogățire a datelor este completat de capacitatea de a dezvolta interfețe de programare a aplicațiilor (API) și de a furniza instrumente de date reutilizabile care pot procesa formate JSON/XML, concentrându-se asupra prezentării lor în browserul web, pentru a oferi o interacțiune „bogată” utilizatorului. Orientarea pe servicii web va continua să contribuie la un schimb de informații mai deschis și mai transparent. Astfel, conectivitatea în IoT trebuie să se bazeze pe formate deschise și nu pe formate de tip software proprietar, deoarece Internetul în sine este un sistem non-propietar care trebuie să ofere utilizatorilor transparență și posibilitatea de a colabora și de a face schimb de informații online.

Un API trebuie să fie suficient de puternic și suficient de clar pentru a satisface nevoile utilizatorilor, cu deschidere către îmbunătățiri viitoare (“durabilitate”). Abordarea API se concentrează pe experiența utilizatorului și, prin urmare, procesul de mixare a informațiilor devine simplu pentru dezvoltator din cauza ușurinței de utilizare și a formatelor simple de transfer, prin *răspândirea informațiilor prin servicii* în loc de pagini statice. Noile tehnologii – precum AJAX (Asynchronous JavaScript and XML) pot implementa servicii Web asincrone care partajează date în format XML sau JSON prin intermediul protocolului HTTP, în conformitate cu paradigma client/server. Modelul orientat spre servicii este simplu, practic și puternic, deoarece complexitatea infrastructurilor IT de bază devine “transparentă” pentru consumatorii de servicii. Un aspect al arhitecturilor orientate pe servicii (SOA) este faptul că facilitează migrarea de la arhitecturile tradiționale IT.

Au fost prezentate un concept și o soluție de analiză web în timp real folosind tehnici standardizate, orientate pe servicii, utilizabile la problemele de analiză a datelor pe baza unor statistici efectuate asupra comportamentului utilizatorilor pe web – un proces continuu (fără feedback – “non interlocked”) care rulează simultan cu progresul navigării. Metadatele de etichetare adăugate măresc valoarea semantică a documentelor web care le conțin (atât din punctul de vedere al editorului, cât și al aplicațiilor concepute să le proceseze).

Motoarele de căutare sunt vitale pentru asigurarea optimă și fiabilă a experienței de interacțiune a utilizatorilor. Soluția propusă susține importanța crescândă a analizei în timp real, utilizarea acesteia în combinație cu învățarea automată fiind promițătoare în creșterea competitivității eBusiness. Chiar dacă au fost remarcate inițial în utilizarea umană, tehnicile de acces potențate de această soluție sunt extensibile în IoT – ceea ce scoate în evidență sustenabilitatea și durabilitatea lor.

3. Abordarea IoT ca Rețea Socială

În ultimii ani, dezvoltarea tehnologică a pus presiune pe mecanismele de schimb de date între diferitele tipuri de dispozitive IoT – fixe și mobile. În acest context, comunicațiile directe între dispozitive, tipice pentru IoT și-au arătat potențialul de a exploata noi aplicații și servicii peer-to-peer bazate pe locație, cu efect de descongestionare a traficului din rețelele tradiționale. Principalele obstacole pentru acest tip de comunicații de la dispozitiv la dispozitiv (D2D - Device-to-Device) sunt eficiența spectrală, latența și precizia [71].

În acest capitol, după extinderea comunicațiilor om-la-om (H2H, human-to-human) la cele *nemijlocite* M2M (machine-to-machine), trecând prin H2M, este abordată *preluarea de către IoT a metodelor din rețele sociale H2H* (bazate pe tehnici colaborative asistate și pe managementul prezenței și disponibilității). Cele mai multe dintre limitările menționate pot fi depășite prin utilizarea noii paradigme "*Internetul Social al Obiectelor*" (SloT – "Social IoT") pentru a ghida în mod autonom relațiile dintre obiectele (și persoanele) coexistente în rețea după regulile pre-stabilite de coexistență și cooperare.

Creșterea constantă a cererii de servicii de date *locale* dar la *rate crescute* a dus la dezvoltarea paradigmei de comunicare de la dispozitiv la dispozitiv (D2D) care permite comunicarea directă a nodurilor mobile din apropiere – din aceeași sub-rețea – fără necesitatea unui controler de nivel superior. Avantajele evidente ale acestui tip de comunicare sunt viteza de transfer mai mare, latența redusă, precizia sporită, eficiența spectrală îmbunătățită și consumul redus de energie [72]. Aceste avantaje se bazează pe mai multe soluții tehnice moderne, printre care tehnici de intermediere (prin "relee"), selecționarea nodurilor, selectarea modului de comunicare, alocarea spectrală a resurselor (bazată pe "spectrum sensing"), gestionarea interferențelor, alocarea canalelor și modernizarea infrastructurilor pre-existente.

3.1. Social IoT

3.1.1. Concept și descrierea sistemului

Noile abordări "sociale" ale comunicațiilor IoT s-au soldat în ultimii ani cu aplicații și arhitecturi de rețea inovatoare care redefinesc stadiul actual al tehnologiei în ceea ce privește *conectivitatea* – de exemplu legăturile wireless, comunicațiile celulare de generația următoare (5G) sau comunicații de la vehicul la vehicul (V2V) sau de la vehicul la infrastructura rutieră [73].

În ultimii ani, ideea convergenței dintre comunicațiile IoT și rețelele sociale a început să atragă atenția sporită a comunității științifice, care a început să aplice soluții din rețelele sociale pentru creșterea performanței comunicării de la dispozitiv la dispozitiv (D2D). Cele mai importante caracteristici sociale care pot fi aplicate comunicațiilor IoT sunt comunitățile sociale, legăturile sociale și conceptul de centralitate. Succesul rapid și difuzarea vastă a unor platforme precum Twitter, Facebook și Instagram, au generat numeroase cunoștințe despre dinamica rețelilor, încurajând cercetătorii din diferite domenii de cercetare să aplice transversal conceptul de "rețea socială" [74] și în domeniul IoT, *folosind aceleași structuri de date, aceleași modele tranzacționale și de management al conexiunii / al sesiunii*.

Rețeaua socială pentru comunicarea în IoT se referă la găsirea și exploatarea tiparelor de interacțiune dintre utilizatori – umani și instrumentali – pentru a spori eficiența rețelilor D2D exploatănd informațiile de proximitate.

3.1.2. Cum pot legăturile sociale îmbunătăți comunicațiile IoT

Comunicațiile D2D au fost introduse inițial pentru a beneficia de:

- proximitatea dispozitivelor pentru asigurarea ratelor de biți mai mari, întârzieri mai mici și consum redus de energie;
- reutilizarea resurselor, permițând simultan atât comunicațiile directe D2D, cât și comunicațiile bazate pe infrastructura rețelelor (de exemplu, celulare sau WiFi);
- utilizarea unei singuri legături între dispozitive, mai degrabă decât implicarea resurselor pentru conexiuni ascendente/descendente.

3.1.3. Descoperirea releelor și selecția nodurilor de tip peer

Atunci când un dispozitiv dorește să stabilească o comunicație D2D, rețeaua și/sau echipamentul de utilizator sursă (UE - *user equipment*) trebuie să găsească care sunt nodurile care pot fi folosite ca releu printr-o procedură de selecție. În ultimii ani, o abordare *social-conștientă* („*socially aware*”) a stimulat dezvoltarea acestei proceduri [76].

Conceptele rețelelor sociale sunt de asemenea utilizate pentru agregarea dispozitivelor în grupuri bazate pe legături sociale (inspirate de cele umane), în care aceste legături sociale sunt folosite ca stimulente pentru a permite comunicarea D2D. Caracteristici sociale din perspectiva rețelei, cum ar fi centralizarea dispozitivelor – adică frecvența legăturilor dispozitivelor cu alte dispozitive și caracteristicile sociale din perspectiva utilizatorului – pot fi folosite pentru transmiterea datelor dar și a reputației, adică a înregistrărilor asupra QoS/QoE (“*quality of service / of experience*”) privind redirectionarea sau retransmiterea informației. O altă caracteristică socială în comunicațiile SLoT este *nivelul de încredere reciprocă* între noduri care contează în abordarea *stop-wait* a procedurii de selectare a releului, astfel încât releul este desemnat prin alegerea legăturii de propagare dar și pe baza nivelului de încredere socială (“*reputație*”). Metrica de încredere este asociată cu istoricul “*socializării*” dintre dispozitive. Figura 10 prezintă arborele de clasificare pentru abordările sociale în selecția nodurilor peer, pe baza celor două aspecte principale abordate în acest context, și anume mobilitatea și încrederea.

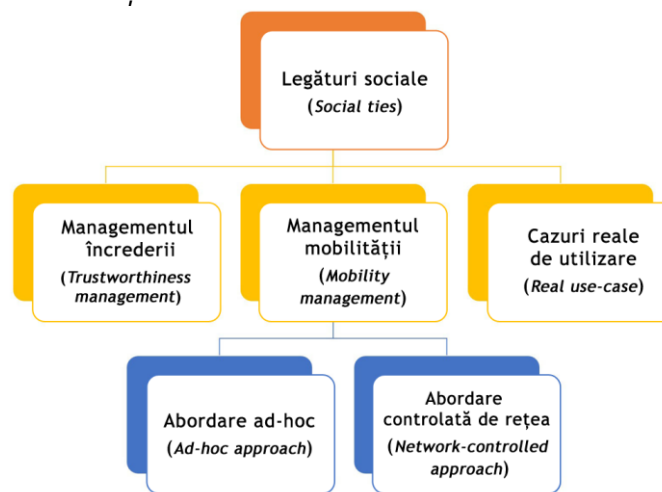


Figura 10: Arborele de clasificare asociat descoperirii releelor și selecției nodurilor de tip peer – abordarea socială

3.1.4. Selectarea modului de comunicare

În timpul procedurii inițiale pentru comunicațiile IoT procesul decizional asupra modului de transmisie este o problemă fundamentală. Pachetele de date sunt transmise printr-o conexiune directă între dispozitivele IoT și, deoarece comunicarea D2D utilizează în mod normal aceeași interfață ca și comunicația celulară, un dispozitiv poate funcționa simultan numai în modul D2D sau în mod celular. Atunci când conexiunile sociale între utilizatori sunt exploatare într-un mod care asigură echitatea, conștientizarea caracteristicilor sociale ar putea fi avantajoasă și pentru calitatea serviciului (QoS) al comunicării cooperative SloT. Alegerea modului de comunicare și problema de selecție a nodurilor peer în etapa de creare a rețelei SloT – identificând utilizatorii cei mai potriviți pentru a se angaja în cooperarea D2D – extrapolează QoS (calitatea serviciilor) către "echitatea" colaborării nodurilor "conștiente social". Disponibilitatea de a coopera – permisiunile de acces acordate altor noduri, acceptul de partajare a propriilor date cu alte dispozitive SloT – beneficiază de creșterea unor rețele sociale precum Twitter sau Facebook și experiența acumulată de acestea va oferi, cu siguranță, oportunități favorabile pentru dezvoltarea comunicațiilor D2D în rețelele 5G [80].

Figura 11 prezintă o prezentare generală a direcțiilor de cercetare actuale pentru selectarea modului de comunicare SloT (pentru fiecare direcție specifică):

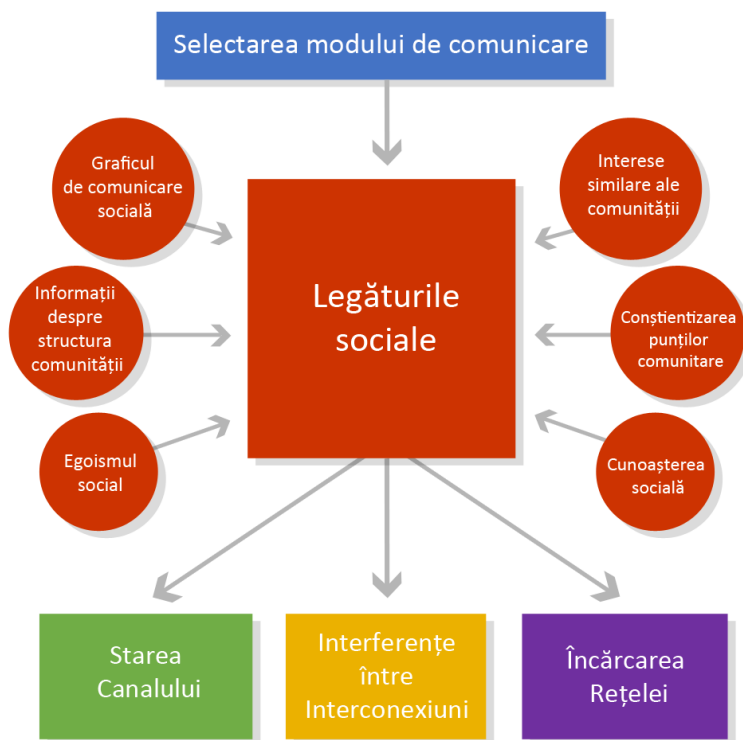


Figura 11: Îmbunătățirea deciziei de selecție a modului de comunicare prin utilizarea informațiilor privind legăturile sociale

3.1.5. Alocarea și gestionarea resurselor

Au fost identificate șase elemente-cheie pentru proiectarea alocării resurselor și optimizarea socială a comunicațiilor IoT:

- Metrici (necesare în decizia automată de alocare) care iau în considerare socializarea dispozitivelor IoT (*"Social-Aware Metrics"*);
- Programarea în timp („*Scheduling*”);
- Heterogenitatea resurselor;
- Optimizarea resurselor;
- Acuratețe de estimare;
- Confidențialitate;

Legăturile sociale dintre dispozitive pot fi folosite ca o metode de învățare automată a comportamentelor de la partenerii cu popularitate / cu reputație sporită. Dispozitivele IoT – fixe sau mobile – pot beneficia de funcții de utilitate bazate pe caracteristici sociale. De exemplu, noii algoritmi pentru căutarea globală urmăresc un echilibrul evolutiv al schimbului de date de control, luând în considerare durata conexiunilor anterioare și similarități ale traiectoriei sociale și ale tipului de comportament (preluând din experiența unor platforme populare, de exemplu Facebook și Twitter) pentru putea să sugereze parteneriate în SloT.

3.2. Web of Things (WoT)

3.2.1. Concept

În IoT s-a urmărit de la început ca – în loc să expună date și funcționalități din lumea reală prin intermediul sistemelor proprietare („închise”) de interfațare – dispozitivele inteligente să poată fi conectate direct la Internet. Rețele de senzori și actuatoare (acționări) fără fir (WSAN – Wireless Sensors & Actuators Networks), dispozitivele cu controller încorporat (*"embedded systems"*), obiectele cu marcaje (tag-uri de identificare) RFID (Radio Frequency IDentification) ș.a.m.d. au căpătat interfețe tipizate la nivelele inferioare OSI – multe din ele Ethernet, pentru legare directă la Internet iar IPv6 le poate alocă adrese pentru fiecare centimetru pătrat din suprafața uscatului planetei. După ce IoT a unificat nivelele inferioare OSI cu Internetul, Web-ul Obiectelor (Web of Things – WoT) cuprinde și unificarea nivelelor superioare OSI cu WWW (World-Wide Web). Așadar conectarea lucrurilor inteligente (și, prin extensie, a obiectelor hardware-firmware-software-netware interconectate) *la Internet prin stratul de rețea*, se continuă prin integrarea IoT cu *Web-ul prin stratul de aplicație* [82], [83].

În capitolul 2 am detaliat stabilirea conectivității IoT într-o varietate de medii de rețea (cu constrângerile respective), iar paragraful următor se va concentra asupra straturilor de prezentare și aplicație – abordările, stilurile arhitecturale software și modelele de programare WoT.

3.2.2. Arhitectură

Dispozitivele WoT sunt adesea *etichetate* (RFID, NFC, coduri QR, coduri de bare etc.), fiind prezente de la WSAN (Wireless Sensors and Actuators Networks) și rețelele vehiculare până la produsele electronice de larg consum [84]. Prin extensie, *obiectele inteligente* și funcționalitatea lor sunt descrise de URI (Uniform Resource Identifiers) – care pot fi transferate, interschimbate și referite pe pagini Web. Obiectele inteligente din WoT pot fi descoperite și agregate prin simpla navigare pe Web.

Arhitectura Web of Things propune patru straturi principale (mai mul în sensul de *etape* și mai puțin în sens OSI) care sunt folosite ca un cadru pentru clasificarea diferitelor modele și protocoale implicate:

- Stratul de accesibilitate („*Accessibility layer*”)
- Stratul de descoperire („*Findability layer*”)
- Stratul de distribuire („*Sharing layer*”)
- Stratul de compoziție („*Composition layer*”).

3.2.3. Metode și tehnologii folosite

REST (Representational State Transfer):

REST este o arhitectură pentru proiectarea aplicațiilor în rețea și o modalitate de a asigura interoperabilitatea între sistemele informatice de pe Internet (a se vedea și paragraful 2.1.2). API-urile construite conform principiilor REST nu cer clientului să știe nimic despre structura API-ului. Mai degrabă, serverul trebuie să furnizeze informațiile prin care clientul să interacționeze cu serviciul.

Metodele HTTP utilizate frecvent într-o arhitectură bazată pe REST sunt.

- GET - Oferă acces la o resursă numai pentru citire;
- PUT - Folosit pentru a crea o nouă resursă;
- DELETE - Este folosit pentru a elimina o resursă;
- POST - Folosit pentru a actualiza o resursă existentă sau pentru a crea o nouă resursă;
- OPTIONS - Folosit pentru a obține operațiile acceptate pe o resursă.

Arhitectura orientată spre servicii (SOA):

SOA (Service-oriented architecture) este o abordare de tip arhitectură pentru definirea, conectarea și integrarea serviciilor reutilizabile, clar delimitate și autonome funcțional [88]. Aceste servicii comunică între ele iar comunicarea poate implica fie transmiterea simplă a datelor, fie poate implica două sau mai multe servicii care coordonează o anumită activitate.

Noile servicii pot fi create pe infrastructura IT existentă a sistemelor SOA, deoarece arhitectura permite reutilizarea activelor existente [89].

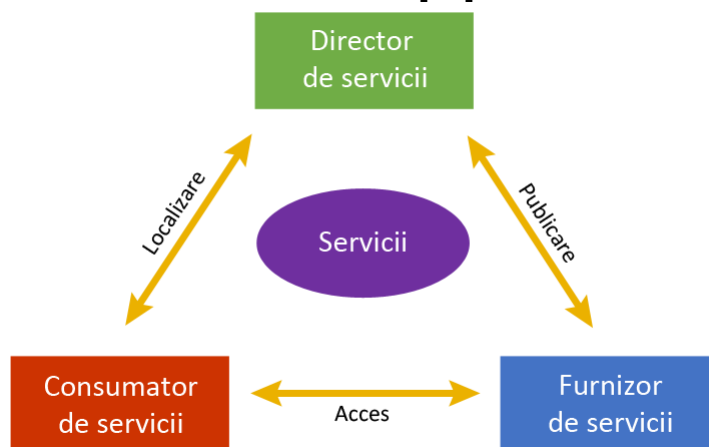


Figura 12: Arhitectura orientată spre servicii (SOA)

3.2.4. Amenințările de securitate din Web of Things

Dispozitivele IoT includ de obicei senzori, comutatoare ș.a.m.d. dar și capabilități de înregistrare care colectează și transmit date pe Internet. Unele dispozitive pot fi utilizate pentru monitorizare, folosind Internetul pentru a furniza actualizări de stare în timp real. Din aceste motive, dispozitivele IoT trebuie să fie protejate împotriva unei game largi de amenințări active și pasive [92].

Amenințări active:

Dispozitivele inteligente pot deveni o amenințare gravă la adresa securității rețelei. Atacatorii pot să utilizeze un dispozitiv IoT compromis pentru a ocoli setările de securitate și pentru a lansa atacuri împotriva altor echipamente de rețea.

Amenințări pasive:

Scopul este de a obține informații detaliate despre obiectiv, mai degrabă decât despre manipularea datelor. Cu toate acestea, atacurile pasive sunt adesea temelia atacurilor active.

Amenințările pasive apar datorită faptului că producătorii colectează și stochează date de utilizator private. Deoarece dispozitivele IoT sunt doar senzori de rețea, aceștia se bazează pe serverele producătorului pentru a face procesarea și analiza datelor.

Mecanisme de securitate în Web of Things (WoT):

Problema securității în mediul conectat rămâne presantă – tehnicile Cloud, mai ales, se dezvoltă efervescent iar unele riscuri de securitate sunt uneori neglijate.

3.3. Metodă de compoziție dinamică a serviciilor colaborative în IoT

3.3.1. Demonstrator SloT - Platformă IoT pentru integrarea și testarea serviciilor colaborative

Obiectivul demonstratorului este de a facilita conexiunea și cooperarea fără întreruperi între dispozitive și utilizatori prin utilizarea unor paradigme moderne precum rețelele sociale și aplicațiile bazate pe IoT.

În cadrul Proiectului de cercetare "RISCOPRIRSI - Design and development of a demonstrator for the dynamic composition of tourism services using an IoT platform" (colaborare într-un program regional italian, sectorul Industrial and Information Engineering - SSD ING - INF/03 - Call sect. 09/F2 - Telecommunications; a se vedea paragraful 7.3.5), am construit această aplicație colaborativă, multi-dispozitiv. Pentru crearea și cadrul de compoziție a serviciilor am extins conceptul de virtualizare a obiectelor fizice astfel încât să se poată adăuga un nou strat de virtualizare și agregare unde *serviciile sunt percepute ca obiecte sociale* și interacționează între ele conform paradigmei SloT.

3.3.2. Arhitectură

Caracteristica principală a demonstratorului este un model care permite utilizatorilor să aibă control complet asupra datelor lor, ceea ce nu este întotdeauna asigurat de soluții alternative. Platforma încurajează, de asemenea, reutilizarea serviciilor și a software-ului dezvoltat de terți și utilizatori. Conceptul de IoT social este completat de stratul de virtualizare. Rezultatul este un agent social autonom, bazat pe o rețea de rețele sociale [93].

Noțiunea de obiecte sociale este folosită într-o arhitectură care permite implementarea de aplicații complet distribuite. Pentru ca cererile să poată fi preluate de aceeași entitate, aceste obiecte sociale sunt implementate în zona Cloud gestionată de utilizatori. În demonstratorul pe care l-am creat, am exploatat modelul PaaS pentru implementarea unei platforme SloT de tip aplicație încrucișată, care să permită clienților să dezvolte, să execute și să gestioneze aplicații fără dificultățile de construire și menținere a infrastructurii asociate. În fig. 17 este prezentată arhitectura de bază a demonstratorului.

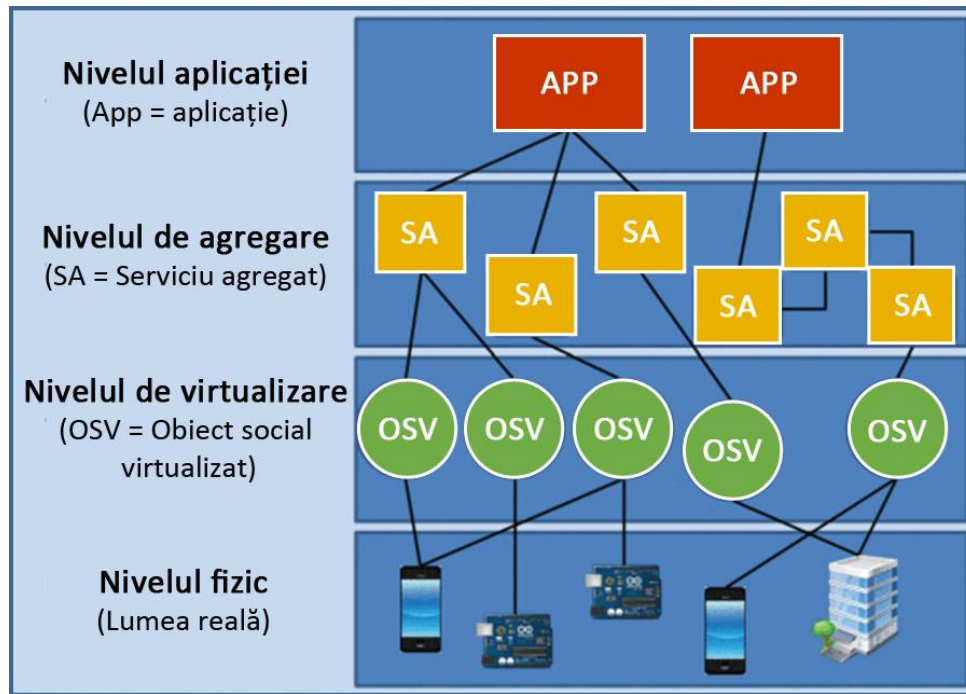


Figura 13: Arhitectura platformei IoT pentru compoziția dinamică a serviciilor conform paradigmei SloT

3.3.3. Nivelul aplicație

La acest nivel, aplicațiile au fost implementate pentru a exploata unul sau mai multe obiecte (agregate). Interfața cu utilizatorul presupune, de asemenea, un rol-cheie – deși soluțiile IoT, sunt centrate pe comunicațiile D2D, centrul de greutate a rămas utilizatorul. Aplicația la acest nivel prezintă o interfață de tip front-end pentru utilizator și o interfață back-end cu straturile care stau la baza acesteia. Procesul de livrare a serviciilor constă în furnizarea *serviciului descoperit*, furnizare condiționată de *serviciul solicitant*.

Odată ce serviciul a fost compus, agentul solicitant (instalat în telefonul în sine) comunică cu agenții care furnizează informațiile relevante pentru a activa serviciul. În tot acest timp, agentul care solicită servicii poate extrage informații importante despre reputația obiectelor care furnizează serviciile. Aceste informații sunt încărcate pe serverul SloT și, prin urmare, sunt disponibile pentru întreaga comunitate. Am construit interfața cu utilizatorul (inclusiv elementele grafice: sigla și logo) ca o aplicație web de tip „Thin Client”, dezvoltată folosind tehnologii moderne, cu un design adaptiv, care, datorită principiilor și tehnicilor receptive (a

se vedea și capitolul 5), oferă o navigație ușor de utilizat și confortabilă chiar și pentru dispozitive diferite, precum se poate vedea în figura 14.

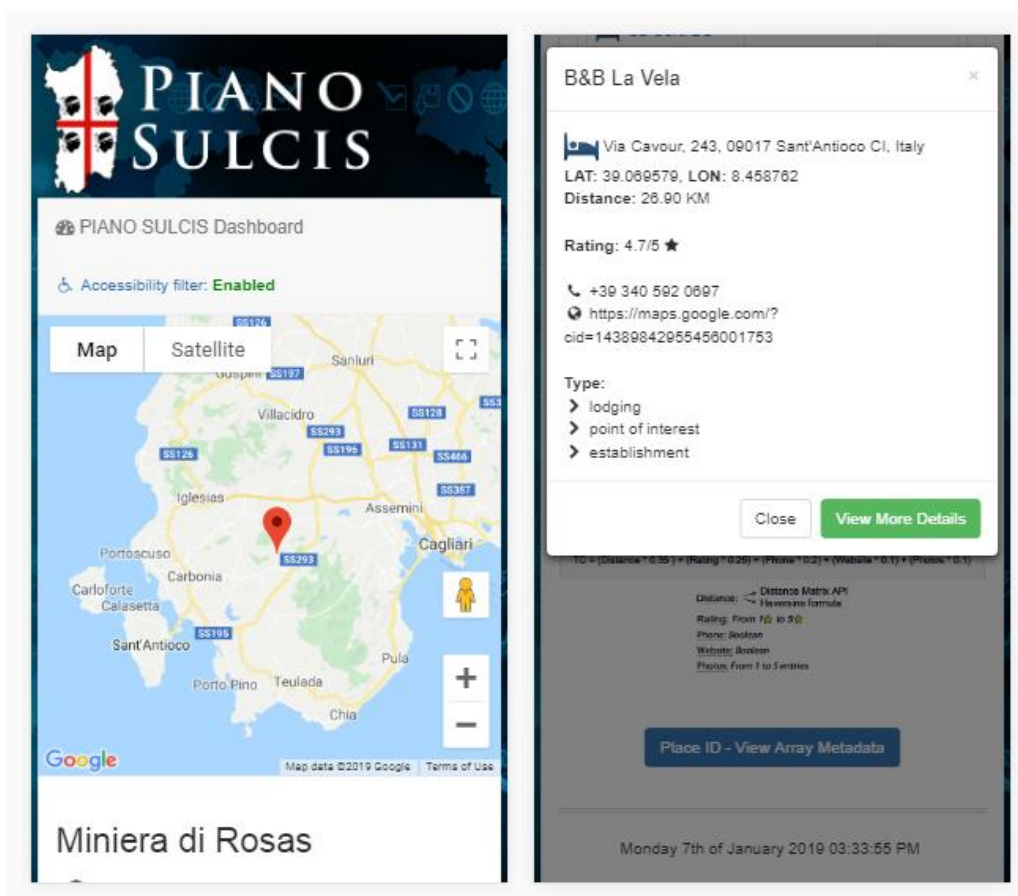


Figura 14: Interfața cu utilizatorul, de tip aplicație web „Thin Client” – smartphone

Astfel, utilizatorul poate consulta „tabloul de bord” al clientului printr-o gamă largă de dispozitive (monitoare de calculator, smartphones, tablete, televizoare etc.) cu același confort vizual și fără a fi nevoie să utilizeze derularea orizontală sau zoom-ul pe dispozitive tactile care, uneori, poate degrada experiența utilizatorului, atât în citire cât și în navigare.

Capacitatea de reactivitate/receptivitate a tabloului de bord este realizată utilizând interogările media CSS3, o extensie a regulii "@media" pentru a adapta aspectul conținutului folosind un sistem bazat pe diferite lățimi, denumite "puncte de rupere" (*breakpoints*) pentru a modifica aspectul, de la un anumit prag critic, în scopul de a ușura afișarea navigării și a conținutului [99]. Pentru a realiza grafice și hărți dinamice am folosit bibliotecile *flot.js*, *morris.js* și *gmap.js*, iar tabelele progresive și interactive au fost construite folosind plugin-ul *DataTables jQuery* [100], precum este prezentat în figurile 15 și 16.

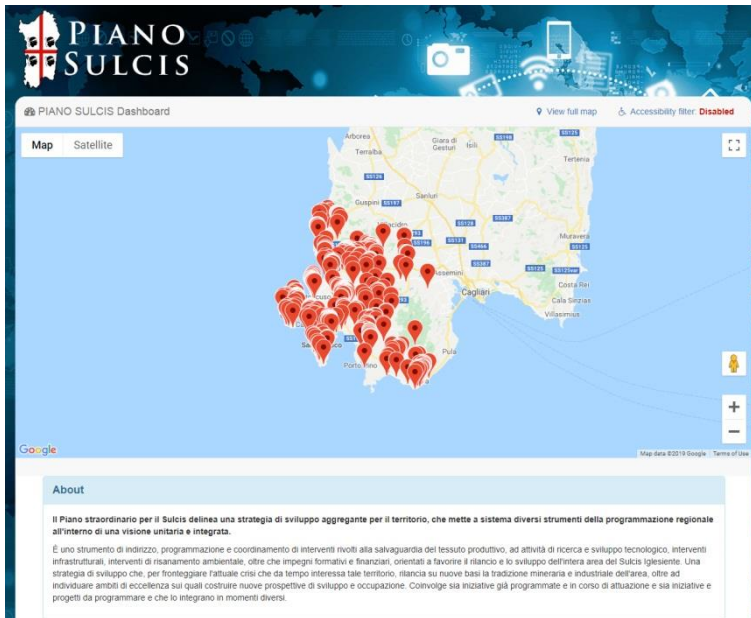


Figura 15: Exemplu de receptivitate a tabloului de bord utilizând utilizând interogări media CSS3

Recommendation Results:

Show 10 entries

Name	Trustworthiness coefficient	Distance (km)
Lu' Hotel	82.5 %	1.46
Agriturismo Sa Scattola	70 %	3.49
FIORIE LEONARDO	68.5 %	27.73
I GIRASOLI TANTISSIMO VACANZE E RELAX	68.5 %	27.78
Villa Comas	68.5 %	29.83
Pizzeria Centrale	68 %	1.47
B&B Plan Di Zuccheru	67.5 %	14.48
Cesù Bù	67 %	14.85
Manku 'S Pub	67 %	2.13
Fano Capo Spartivento	67 %	41.65

Showing 1 to 10 of 197 entries

Previous 1 2 3 4 5 20 Next

Trustworthiness Coefficient (TC)

$$TC = (Distance * 0.35) + (Rating * 0.25) + (Phone * 0.2) + (Website * 0.1) + (Photos * 0.1)$$

Distance: Distance Matrix API Haversine formula
 Rating: From 1★ to 5★
 Phone: Boolean
 Website: Boolean
 Photos: From 1 to 5 entries

Figura 16: Exemplu de tabel progresiv și interactiv construit folosind plugin-ul *DataTables jQuery*

3.4. Sumarul capitolului

Scopul prezentului capitol a fost de a explora și de a înțelege beneficiile integrării metodelor din rețele sociale cu comunicarea IoT pentru a oferi o imagine clară a eforturilor științifice actuale din acest domeniu și pentru a crea o taxonomie a aplicațiilor identificate.

În concluzie, selecția modului de comunicare și de interacțiune între obiectele IoT poate include diferite legături sociale care vor fi utilizate pentru condiționarea canalelor de transmisiuni, gestionarea interferențelor și a încărcării rețelei, pe baza unor concepte sociale precum: interese comunitare, mediere, cunoaștere socială, altruism social și informații despre structura comunității.

Combinarea schemelor de descoperire a rețelelor și a modurilor de comunicare pentru stabilirea relațiilor sociale poate fi utilizată atunci când conexiunile sociale între utilizatori sunt exploatate într-un mod care asigură etica și echitatea ca extensie socială a calității serviciilor (QoS) / a calității experienței (QoE). Perspectiva selecției modului de comunicare bazată pe conștientizare socială ("social awareness") va fi probabil imprevizibilă în dezvoltarea viitoare și e nevoie în continuare de o mai bună înțelegere a relațiilor sociale în vederea selecției automate optimale dintr-un grup de caracteristici sociale axate pe informația despre structura comunității și pe similitudinile de intereselor.

Ulterior am descris arhitectura și conceptul Web of Things și cele mai bune practici bazate pe principiile RESTful care contribuie la scalabilitatea și modularitatea WoT.

În final am descris contribuția la implementarea serviciilor ca obiecte sociale virtualizate și am prezentat demonstratorul SIoT realizat, care prezintă patru caracteristici majore: 1) a fost conceput pentru a exploata modelul de serviciu PaaS; 2) a adăugat un nou strat de virtualizare peste cel al obiectelor fizice – acela de servicii ca și obiecte sociale; 3) datele și aplicațiile utilizatorilor sunt stocate și executate în spațiul Cloud al utilizatorului; 4) reutilizabilitatea șabloanelor și a aplicațiilor.

4. Integrarea IoT în Cloud prin comunicații web de timp real (Web-RTC)

4.1. Comunicații web de timp real

WebRTC (prescurtarea pentru Web Real-Time Communications) este o colecție de protocoale de comunicații și interfețe de programare a aplicațiilor (API-s) care permit comunicarea în timp real (cu o latență care a coborât mult sub constantele de timp ale percepției umane sau ale proceselor industriale) folosind conexiuni "peer-to-peer" (P2P) [101]. Acest lucru permite navigatoarelor web să trimită și să primească informații în timp real, din și către navigatoarele altor utilizatori – premisa pentru aceasta o constituie noile performanțe de viteză și capacitate ale comunicațiilor IP.

4.1.1. Arhitectura WebRTC

Modelele web tradiționale sunt dezvoltate pe arhitectura client-server în care comunicarea se realizează unidirecțional (de la server la client prin protocolul HTTP). Arhitectura WebRTC dezvoltă modelul client-server prin înglobarea noțiunii de comunicare P2P ("peer-to-peer") între navigatoarele web [103]. Cea mai folosită arhitectură WebRTC este modelul triunghiular în care cele două navigatoare web accesează o aplicație pe un server (aflat "în vârful triunghiului") – prin intermediul acestei aplicații, mesajele de semnalizare sunt transmise prin protocolul HTTP sau tehnologii WebSocket [104]. Serverul web folosește doar pentru a controla transferul de date, însă transferul de volum mare al fluxurilor efective (streaming) de date este realizat "peer-to-peer" între navigatoare (browsere).

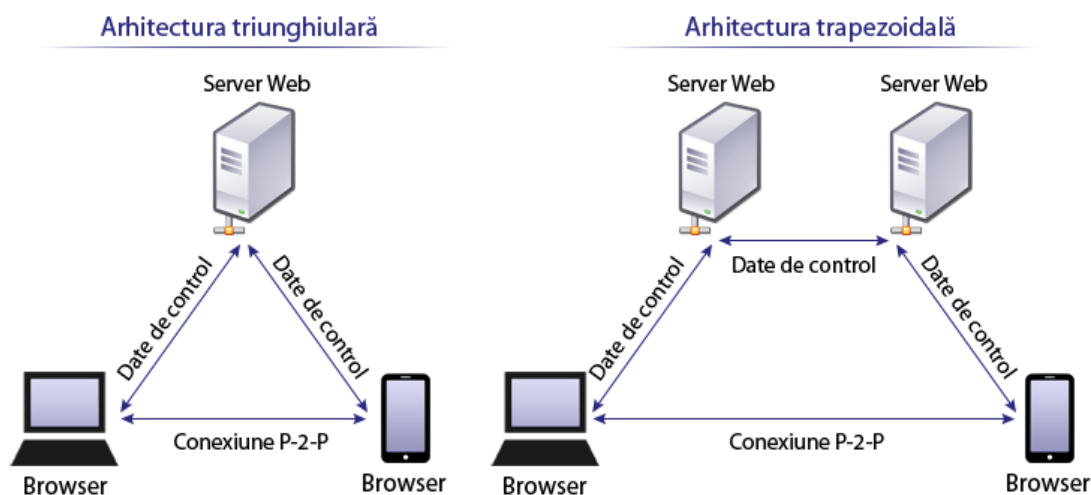


Figura 17: Arhitectura WebRTC – modelul triunghiular (și cel "trapezoidal", cu două servere)

4.1.2. API WebRTC

API-ul WebRTC este proiectat pentru a permite aplicațiilor JavaScript să creeze conexiuni în timp real folosind canale audio, video și/sau de date direct între utilizatori prin intermediul navigatoarelor web, sau serverelor care susțin protocoalele WebRTC. De asemenea, API-ul WebRTC poate folosi metoda "Media Capture" `getUserMedia()` pentru a obține acces la microfonul sau camera web instalate pe terminalul utilizatorului [105].

API-ul WebRTC este construit în jurul a trei concepte:

- **RTCPeerConnection** - sarcina principală a obiectului *RTCPeerConnection* este de configurare și de a crea o conexiune de la egal la egal (peer-to-peer);
- **MediaStream** - oferă acces la obiecte de tip fluxuri video și audio;
- **RTCDataChannel** - creează un canal care vine dintr-un obiect *RTCPeerConnection* existent.

MediaStream API

API-ul MediaStream a fost proiectat pentru accesul facil la fluxurile media de la camerele web și microfoanele locale. Metoda `getUserMedia()` este principalul mod de a avea acces la dispozitivele de intrare locale [106].

API-ul Media Stream are trei caracteristici principale:

- Un flux media în timp real este reprezentat de un *obiect flux* – în sensul IoMT (Internet of Multimedia Things) – sub formă video sau audio;
- Ca măsură de securitate, este solicitat acceptul utilizatorului înainte ca o aplicație web să poată porni preluarea unui flux;
- Selectarea dispozitivelor de intrare este manipulată de către API MediaStream (de exemplu, atunci când există două camere sau microfoane conectate la dispozitiv).

În extensia *Firebug* a navigatorului se poate observa cum API-ul WebRTC a generat un URL de tip *blob* („*Binary Large Object*”) pentru obiectul *MediaStream* (în aceeași perspectivă IoMT).

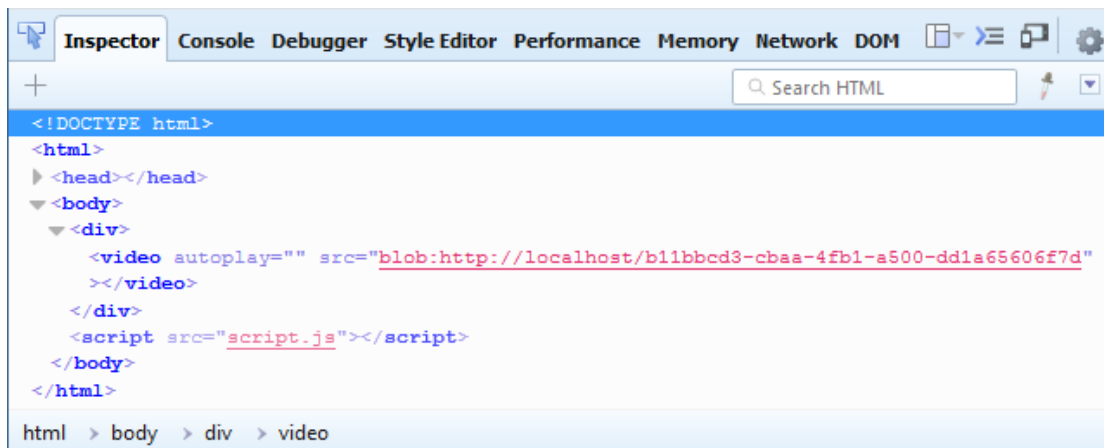


Figura 18: Exemplu de URL de tip *blob* pentru obiectul MediaStream

4.1.3. Protocoalele STUN și TURN

STUN (Session Traversal Utilities for NAT) este un set standardizat de metode, inclusiv un protocol de rețea, pentru traversarea dispozitivelor NAT – Network Address Translation (Traducerea Adreselor de Rețea) în aplicațiile de voce în timp real, video, mesagerie, precum și alte comunicări interactive [100]. STUN este un instrument folosit de alte protocoale, cum ar fi ICE (menționat anterior) sau protocolul Session Initiation Protocol (SIP). Acesta oferă un instrument pentru descoperirea prezenței unui IP în interiorul unei rețele care folosește un translator de adrese de rețea. Protocolul necesită asistență de la un server de terță parte (server STUN), situat de obicei public pe Internet.

TURN (Traversal Using Relays around NAT) este un protocol folosit în ocolirea translatoarelor de adrese de rețea (NAT) și a *firewalls* în scenarii multimedia [111]. Acesta poate fi utilizat împreună cu protocolul de control al transmisiei (TCP - Transmission Control Protocol) și User Datagram Protocol (UDP). Este folosit în mod special pentru utilizatorii de pe rețelele cu dispozitive NAT simetrice și gestionează conexiunea unui utilizator în spatele unui NAT către un singur nod (peer).

Tehnologia ICE permite browserelor să descopere suficiente informații despre topologia rețelei în care acestea sunt utilizate pentru a găsi cea mai bună cale de comunicare [112]. ICE oferă, de asemenea, o măsură de securitate, deoarece împiedică paginile web și aplicațiile nesigure să trimită date către gazde ("hosts") care nu se așteaptă să le primească.

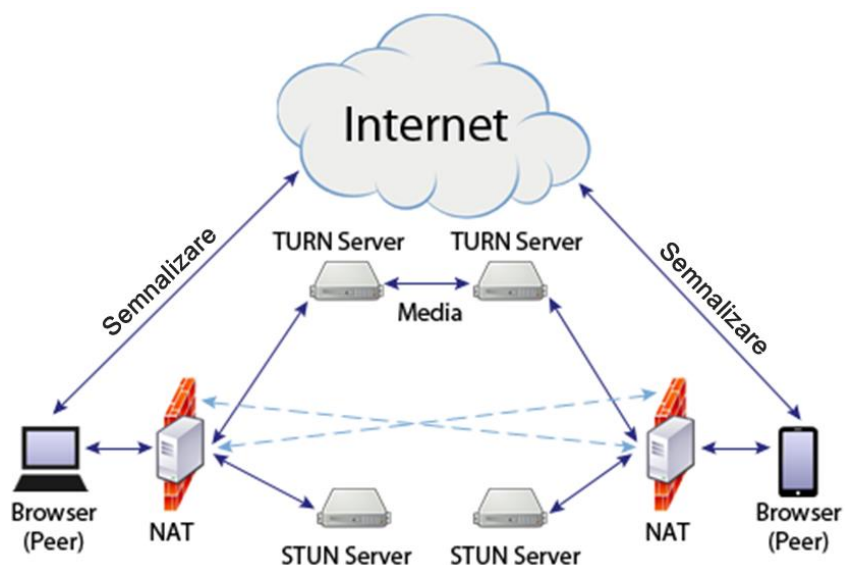


Figura 19: ICE - Interactivity Connection Establishment folosind STUN și TURN

4.1.4. Semnalizarea în WebRTC

Pentru ca un utilizator să se conecteze la un alt utilizator trebuie să cunoască unde se află acesta pe Web. Adresa IP permite dispozitivelor conectate la Internet să trimită date direct între acestea.

În comunicațiile web în timp real rolul semnalizării este de a ajuta dispozitivele să se găsească reciproc pe Internet pentru a putea face schimb de date cu privire la protocoalele și codec-urile acceptate de fiecare dintre ele [113].

Semnalizarea este procesul esențial de coordonare a comunicațiilor. Pentru ca o aplicație WebRTC să poată configura un "apel", clienții trebuie să poată face schimb de următoarele informații:

- mesaje de control al sesiunii utilizate pentru a deschide sau închide o comunicație;
- mesaje de eroare;
- metadate media, cum ar fi codec-uri și setările acestora, lățimea de bandă și tipurile de fișiere media folosite;
- date cheie, utilizate pentru a stabili conexiuni securizate;
- date de rețea, cum ar fi adresa IP a gazdei și portul de ieșire în Internet.

Pentru a evita redundanța și pentru a maximiza compatibilitatea cu tehnologiile emergente, metodele și protocoalele de semnalizare nu sunt specificate de către standardele WebRTC, fiecare dezvoltator fiind liber să folosească tehnica dorită.

4.1.5. Avantaje și motive pentru a folosi tehnologia WebRTC

Tehnologia WebRTC poate contribui la integrarea IP a ecosistemelor IoT / IoMT ridicând, în mare măsură, metodele de interconectarea rapidă către partea superioară a stivei OSI. Enumerarea de mai jos cuprinde principalele avantaje ale tehnologiei WebRTC menționate în literatura de specialitate [114]:

- Este independentă de dispozitive și platforme;
- Transfer de date audio/video securizat;
- Calitate avansată audio și video;
- Sistem de încredere pentru stabilirea sesiunii ;
- Fluxuri media multiple;
- Adaptarea la condițiile de rețea;
- Interoperabilitatea cu VoIP;
- Dezvoltarea rapidă a aplicațiilor;
- Comunicații încorporate ("*Embedded communications*");
- Viruși, Malware, Ransomware și Spionaj industrial;

4.1.6. Biblioteci și seturi de instrumente WebRTC

Următoarea listă prezintă principalele biblioteci și seturi de instrumente, pentru utilizarea optimizată a tehnologiei WebRTC, menționate în literatura de specialitate [115]:

- **PeerJS** este o bibliotecă care adaptează și sporește integrarea WebRTC în browser pentru a oferi un API complet configurabil și ușor de utilizat pentru conexiunile peer-to-peer;
- **EasyRTC** este un set de instrumente WebRTC adecvat pentru construirea de aplicații fiabile și de înaltă securitate. Este un pachet de aplicații web, fragmente de cod, biblioteci și componente de server, scrise și documentate pentru acces și dezvoltare facilă.
- **adapter.js** este o bibliotecă care, pentru fiecare versiune a fiecărui browser care acceptă WebRTC, implementează blocurile de cod necesare, stabilește rădăcinile denumirilor de APIs și aplică modificările necesare pentru a face browser-ul să ruleze corespunzător codul sursă conform specificațiilor WebRTC.
- **simpleWebRTC** este o bibliotecă compusă din mai multe module independente.

- **Socket.IO** este o bibliotecă JavaScript care asigură comunicarea bidirecțională în timp real bazată pe evenimente. Este o bibliotecă multi-platformă, multi-browser și multi-dispozitiv, concentrându-se în mod egal asupra fiabilității și vitezei;
- **ShareFest** este un instrument construit pe canalele de date ale API-ului WebRTC folosit pentru partajarea peer-to-peer a fișierelor;
- **Hoodie** este un set de instrumente pentru o abordare simplificată (doar la "front-end" – practic fără "back-end") bazată pe metodele deja încorporate în browser (așadar "offline-first" în prioritizarea fiecărei cereri și, implicit, cu o trecere imperceptibilă de la online la offline). Hoodie e scrisă în JavaScript și NodeJS, fiind bazată pe CouchDB ; oferă o serie de unelte precum stocare de date, înregistrare și administrare de utilizatori, e-mail-uri, plăți online sau sincronizări.

4.2. Contribuție la comunicarea peer-to-peer în timp real

Acest subcapitol este dedicat implementării unei soluții de comunicare web în timp real utilizând WebRTC, Unify Circuit, Node.js, Express Framework și OpenSSL. Aceste noi tehnologii [116] au fost integrate într-o aplicație care este complexă constructiv dar are avantajul simplității în utilizare.

4.2.1. Mediul de dezvoltare și mediul de control al apelurilor

Mediul de dezvoltare al aplicațiilor este extins, cuprinzând instalarea de servicii de dezvoltare software și platforme software pentru securitate și criptare, cum ar fi Java SDK (kitul de dezvoltare de software Java), Node.js, Python, Circuit și OpenSSL.

Modulul de control al apelurilor: în Node.js, toate modulele exportă elemente care pot fi ușor apelate oriunde în cod, iar pentru logare este utilizată o instanță a modului *Express*. *Express* este un modul rapid de dezvoltare web care oferă diferite funcții pentru vizualizare, testare, rutare, dar și pentru generarea de aplicații executabile. Funcția *logon()* compară e-mailul de conectare și parola cu baza de date de pe platforma *Unify Circuit*, apoi creează o instanță de client și prin intermediul metodei *events* permite accesul utilizatorului în aplicație.

```

_client.logon($email.value, $password.value).then(function(user) {
  _localUser = user;
  $email.disabled = true; //default filled field
  $password.disabled = true; //default filled field
  $login_wall.style.display = 'none';
  $content.style.display = '';
  $logonButton.style.display = 'none';
  document.getElementById('name').innerHTML = user.displayName;
  document.getElementById('right_side').style.display = '';
})

```

Figura 20: Funcția *logon()* - creează o instanță de client prin intermediul metodei *events*

Funcția care modifică starea conexiunii aplicației utilizează metoda *addEventListener()*, atașând un eveniment la elementul Client, fără a suprascrive alte evenimente deja create, precum este prezentat în cele ce urmează.

```

function addClientEventListeners() { //adding a client
  if (_client) {
    _client.addListener('connectionStateChanged',
      function onConnectionStateChanged(evt) { //change status
        //adding "connected" or "disconnected"
        if (evt.state === Circuit.Enums.ConnectionState.Disconnected) {
          //in case of disconnection returns to initial interface
          resetCallUI(); //interface reset
        }
      }
    );
  }
};

```

Figura 21: Funcția *addClientEventListeners()* - modifică starea conexiunii aplicației atașând un eveniment la elementul Client

4.2.2. Gestionarea conexiunii

Managementul conexiunii: pentru a realiza conexiunea între cele două entități, se setează stratul de legături de date: *path* și *app.js*. Funcția *app.set*, de tip (*nume, valoare*) stabilește numele și valoarea portului local în care rulează aplicația.

Subprogramul *index.js* conține funcția cerere-răspuns care exportă pagina web care include aplicația. De asemenea se generează serverul virtual care găzduiește aplicația (ex: *https://localhost:8080*).

```

//creating virtual server
var httpsServer = https.createServer(credentials, app);
httpsServer.listen(app.get('port'));

```

Figura 22: Generarea serverului virtual care găzduiește aplicația

Un alt aspect de conectare este adăugarea tag-ului *<script>* care conține elementele de sub-interfață *index.ejs*. Etichetele se referă fie la un link, la un tip de document script, fie la un fișier *client.js* local. Codul din figura următoare conectează interfața *index.ejs* la toate subprogramele programului principal numit *client.js*.

```

<script src='https://circuitsandbox.net/circuit.js'></script>
<script src='js/client.js'></script>

```

Figura 23: Conectarea interfeței *index.ejs* la toate subprogramele programului principal

4.2.3. Gestionarea fluxului

Managementul fluxului: funcția *start()* verifică mai întâi dacă utilizatorul este logat și creează variabile de timp, apoi este selectată o conversație iar streaming-ul media este pornit. Funcția *join()* verifică dacă utilizatorul este conectat, creează variabile pentru tipurile de suporturi media și permite utilizatorului să aleagă o conversație [117]. Pentru a participa la o conversație, funcția trebuie să aibă starea "started".

```
function updateList(call) { //updates conversations list
  if (call.state === Circuit.Enums.CallStateName.Terminated) {
    //if the conference ends
    getListOption(call.convId).textContent =
      _conversations[call.convId].title;
  } else {
    getListOption(call.convId).textContent =
      _conversations[call.convId].title + '(' + call.state + ')';
  }
  //if not ended further display the status
}
```

Figura 24: Gestionarea fluxului și schimbarea stării prin intermediul funcției *updateList()*

4.2.4. Interfața grafică cu utilizatorul

Interfața grafică cu utilizatorul (GUI): O funcție *setCallUI()* este creată și utilizată de fiecare dată când se lansează configurarea sau resetarea interfeței cu utilizatorul [118]. În următoarea funcție, butonul care accesează conferința este schimbat. Acesta are două opțiuni: "Start" - care se referă la începutul unei conferințe și "Enter" - care se referă la participarea la o conferință care a început deja.

Funcția *updateButtons()*, prezentată mai jos, se referă la butonul care își schimbă textul în funcție de starea conversației. Dacă conferința a început deja, funcția va afișa "Join" iar dacă un utilizator inițiază conferința, funcția va afișa "Start". Când conferința nu este pornită, funcția va ascunde butonul "Exit conference".

```
function updateButtons() {
  $startButton.classList.add('hide'); //hide "Start" button
  $joinButton.classList.add('hide'); //hide "Join" button
  $leaveButton.classList.add('hide'); //hide "Leave" button
  if (!call) { //If user has not yet entered a conference

    $startButton.classList.remove('hide'); //hide "Start" button
  } else if (call.state === Circuit.Enums.CallStateName.Started) {
    //if a conversation is already "Started"
    $joinButton.classList.remove('hide'); //hide "Join" button
  } else if (call.state === Circuit.Enums.CallStateName.ActiveRemote) {
    $leaveButton.classList.remove('hide'); //hide "Leave" button }
}
```

Figura 25: Gestiunea controalelor stării conversației: "Join", "Start", "Exit conference" prin intermediul funcției *updateButtons()*

4.3. Contribuție la autentificarea în Cloud prin procesarea obiectelor multimedia

În decursul ultimului deceniu, autentificarea folosind date biometrice, cu precădere recunoașterea facială, a devenit o zonă populară de cercetare iar paradigma *Cloud computing* a câștigat tot mai mult teren, asigurând astfel resursele necesare pentru analiza imaginilor în timp real dezvoltându-se algoritmi din ce în ce mai noi și mai puternici de detectare a feței [119]. Perspectiva utilizării soluției cercetate este orientată mai mult pe testarea inteligență bazată pe comunicații în timp real, cu „clienți subțiri” și afișare a fețelor încorporate în browserele obișnuite, dar cu analize puternice de date în Cloud, pentru autentificare sau clasificare. Așadar, în contextul mai general al lucrării, metodologia dezvoltată se poate extinde la alte soluții IoMT bazate pe WebRTC.

4.3.1. Tehnologii și implementare

Soluțiile prezentate sunt bazate pe Cloud, folosind "clienți subțiri – Thin Clients" pentru tablete sau smartphones și fără dependențe computaționale localizate - de exemplu, simplul streaming al camerei web spre afișare în browserele obișnuite prin WebRTC lasă ca toate procesele de fundal pentru analiza datelor (folosind tehnici ale inteligenței artificiale – Machine Learning/Deep Learning [123]) să ruleze în Cloud. Recunoașterea facială este implementată cu metode de rețea neuronală Tensorflow/OpenFace/Microsoft Azure Cognitive Services, utilizând Python, Torch, OpenCV și Face API [124].

4.3.2. Abordarea fluxului de lucru pe rețele neuronale

OpenFace este o soluție de recunoaștere a feței de tip open-source care utilizează rețele neuronale implementate în Torch și Python dezvoltate la Universitatea Carnegie Mellon din Pittsburgh, Pennsylvania, SUA [125].

Prezentarea generală a fluxului de lucru OpenFace:

1. Detectarea facială utilizând un model pre-instruit (OpenCV);
2. Transformarea feței pentru rețelele neuronale;
3. Reprezentarea feței pe o hipersferă 128-dimensională utilizând rețele neuronale profunde („*deep neural networks*”).

Unul dintre principalele avantaje ale utilizării Python în dezvoltarea de aplicații este existența bibliotecilor extinse pentru metode standard. Captura video și transferul media în timp real s-au realizat utilizând tehnologia WebRTC [127]. Setul de date privind antrenamentul pentru recunoașterea feței este rezident în Cloud și poate fi procesat cu ajutorul bibliotecii software TensorFlow pentru învățare asistată (machine learning) sau a altor sisteme și algoritmi de rețele neuronale de deep learning (“învățare profundă”). Pentru recunoașterea facială pot fi folosite imaginile din actul de identitate (ori ecusonul firmei etc) sau un set de imagini furnizate și validate în prealabil. Pentru cele mai bune rezultate, este recomandat să existe mai multe imagini pentru fiecare subiect din setul de antrenament.

4.3.3. Abordarea prin servicii integrate Cloud

Microsoft Azure este o colecție de servicii integrate Cloud concepute pentru a construi, implementa și gestiona aplicații și servere printr-o rețea globală de centre de date. Printre cele mai importante servicii oferite se numără: serviciile de calcul (pe mașini virtuale, servicii de aplicații, situri Web), servicii mobile, servicii de stocare, învățare automată, gestionarea datelor, etc [128].

Unul dintre cele mai remarcabile avantaje ale Microsoft Face API este că, pe lângă recunoașterea reală a feței, poate oferi și informații valoroase despre emoțiile subiectului și poate organiza chipurile împreună în grupuri, pe baza asemănării lor vizuale. Procesul de dezvoltare și implementare a unei astfel de aplicații este relativ simplu și direct, deoarece este în întregime bazat pe Cloud. Trebuie să se creeze un cont Microsoft Azure și, din tabloul de bord, să se selecteze tipul de aplicație și resursele necesare [130]. În figura de mai jos, este prezentată caracteristica de detectare a feței. În ambele cazuri, aplicația determină, de asemenea, genul și emoția subiectului (zâmbind sau nu). Folosind algoritmi complecși, aplicația se apropie, de asemenea, de vârsta subiectului, bazându-se pe atributele feței.

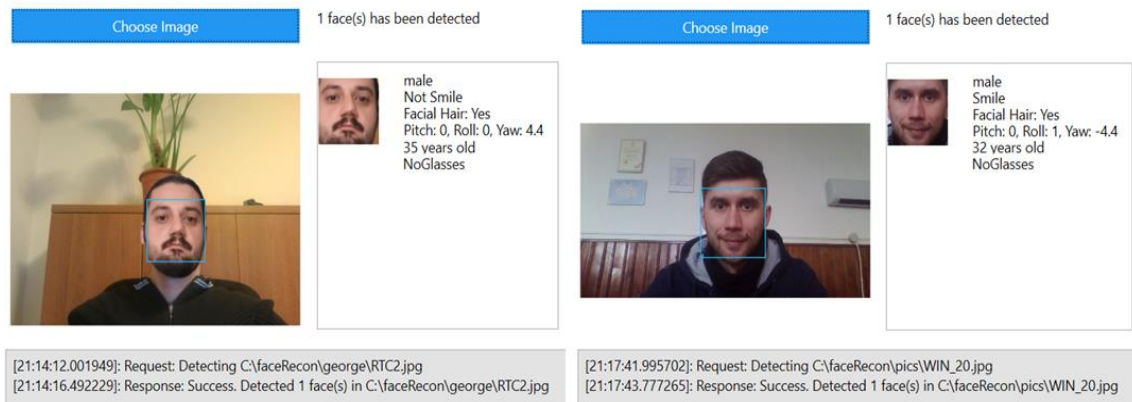


Figura 26: Exemplu de detectare a feței

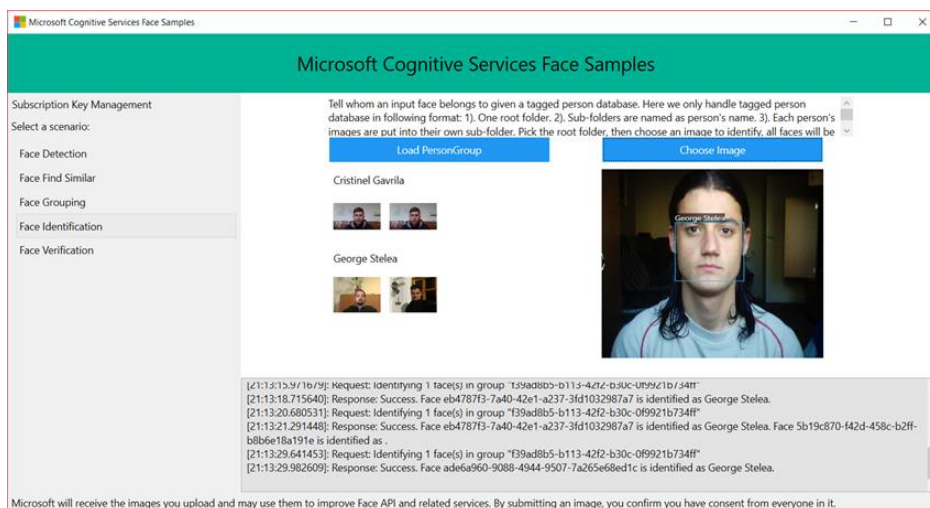


Figura 27: Exemplu de identificare a feței

Imaginea de mai sus prezintă caracteristica de identificare a feței. Pentru această caracteristică, aplicația are nevoie de câteva fotografii anterioare care sunt folosite ca referință.

Versatilitatea API-ului asigură identificarea reușită chiar dacă imaginile subiectului au fost realizate în cadre diferite, așa cum este cazul prezentat, unde vârsta subiectului din imaginea de referință este semnificativ mai mare decât cea utilizată pentru identificare.

4.4. Sumarul capitolului

Utilizarea tehnologiei WebRTC în IoT / IoMT reduce semnificativ resursele fizice necesare (porturi fizice de intrare/ieșire și infrastructura de interconectare), resursele de memorie (de lucru și de stocare) și necesitățile de întreținere a software-ului: actualizări ale sistemului de operare, instalarea sau deinstalarea de module software și chiar manevrele de securizare, monitorizare și control. Un factor de succes WebRTC este acela că tehnologiile sale de bază (HTML, HTTP, TCP/IP) sunt deschise și ușor integrabile. Deja integrat cu cele mai bune soluții co-dec pentru voce și video, WebRTC include și tehnologii netware avansate de traversare a firewalls, folosind STUN (Session Traversal Utilities for NAT), conectivitate interactivă și suport RTP-over-TCP pentru proxy. De asemenea, dezvoltatorul poate folosi orice – XMPP, Jingle sau SIP etc – în protocol în funcție de scenariul său de exploatare. Așa cum ilustrează acest capitol, WebRTC poate fi integrat și ajustat pentru orice client web. În prezent, unele dintre cele mai mari companii IT din lume - Google, Amazon, Oracle, Facebook, Twilio, Citrix etc - au implementat deja această tehnologie în aplicațiile lor. Acest lucru evidențiază proliferarea dezvoltării și integrării WebRTC în aplicațiile web și orientarea către Cloud a comunicațiilor digitale.

API-urile și standardele WebRTC vor "democratiza" și descentraliza instrumentele pentru crearea de conținut și comunicare - pentru telefonie, jocuri, producție video, producție de muzică, colectarea știrilor și multe alte aplicații. WebRTC va avea un impact deosebit asupra tehnologiilor web deschise și interoperabile ale browser-ului, inclusiv a soluțiilor *enterprise* existente.

Soluțiile și contribuțiile prezentate au un spectru larg de aplicabilitate în IoT / IoMT, de la asigurarea accesului securizat până la transmiterea în timp real a obiectelor multimedia prin comunicații Cloud.

Am implementat o soluție de recunoaștere a feței care poate fi integrată într-un sistem de management al învățării (LMS - Learning Management System) și poate fi utilizată atât în învățământul universitar, cât și în alte medii educaționale. Din perspectiva paradigmei "Educație ca serviciu", soluțiile de autentificare bazate pe WebRTC pot asigura un mediu favorabil pentru examenele online sau pentru seminarele web în privința autentificării și a interactivității sporite.

Valoarea de întrebuințare a acestor soluții și tehnologii e dată de potențialul de extindere a metodologiei de dezvoltare și la alte soluții IoMT bazate pe WebRTC, dată fiind și evaluarea calculabilității (prin aceste studii de caz care necesită distribuția judicioasă în Cloud a unor procese intensive computațional cum ar fi învățarea asistată și clasificarea prin metode ale inteligenței artificiale).

5. Servicii Web pentru Telematică în IoT

În informatica instrumentală tendința modernă este către o transpunere completă a tuturor conexiunilor (nu numai pe Internet, ci și local în cadrul instrumentelor virtuale) și a tuturor schimburilor de date și comenzi ca simple mesaje HTTP. Astfel, o singură structură de tranzacție și aceeași logică de servicii (inclusiv automatizarea integrării în Cloud) pot fi implementate independent de distanță și / sau poziție [131].

Decizia de "instanțiere" pentru alocarea resurselor, inclusiv gestionarea resurselor ca serviciu, poate fi "fără stare" / "fără conexiune" / "fără server" ("stateless" / "connectionless" / "serverless"), cu noi criterii de optimizare (în principal eficiența costurilor) – păstrând totuși preocuparea pentru o securitate sporită [132].

Așa cum am arătat, serviciile web pentru tele-măsurare sunt abordate conform setului de recomandări REST (Representational State Transfer) cu reprezentarea unificată *stare* – *tranzicție* / *evenimente* ce permite integrarea managementului resurselor în mediul distribuit prin intermediul "mașinilor algoritmice de stare" ("automate cu stări finite") asociate modelelor de comportament / de control ale instrumentelor [133].

Datele produse de senzori, însoțite de metodele lor de prezentare, sunt preluate de un "Thin Client" web – un "tablou de bord" ("control board" / "dashboard") dinamic, cu adaptarea conținutului la terminal – dezvoltat pentru a gestiona și a conecta serviciile de telematică, utilizând o interfață grafică cu utilizatorul, cu acces simplu bazat pe browsere obișnuite și fără a fi nevoie să se instaleze module ("plugins") terțe suplimentare.

5.1. Contribuții la tele-măsurare folosind instrumentație virtuală și servicii web

În serviciile web REST, mecanismele simplificate de răspuns la cereri sunt construite pentru a gestiona în Internet resursele adecvate de creare-citire-actualizare-ștergere ("CRUD" – create-read-update-delete) [134].

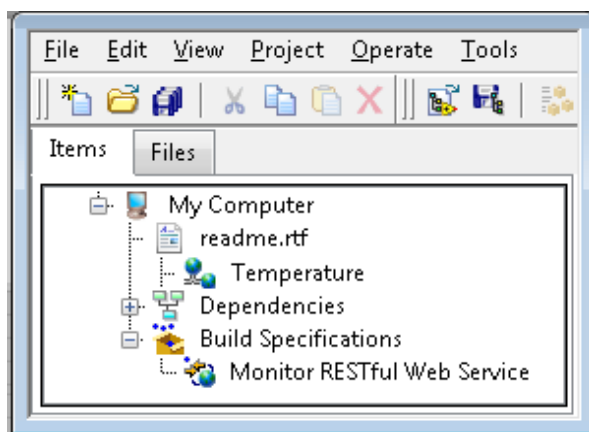


Figura 28: Exemplu de suport National Instruments pentru serviciile RESTful – într-un configurator de proiect LabVIEW, "Temperatura" se poate observa ca variabilă partajată pe Web

Aceste acțiuni reprezentate în HTTP au un avantaj esențial (inclusiv sub aspectul securității IP): *se reduce la minim numărul de porturi* care trebuie deschise pentru comunicațiile industriale integrate cu IoT.

5.1.1. Transmiterea datelor de la senzori și traductoare inteligente

Pentru tele-măsurare în Cloud am implementat un demonstrator bazat pe ESP8266, produs de Espressif Systems, un chip Wi-Fi rentabil și foarte popular, cu o stivă completă TCP / IP integrată, guvernată de un microsystem RISC încorporat într-un controller pe 32 de biți -TensilicaXtensa L106 care funcționează la 80 MHz, pe baza instrucțiunilor stocate în 64 KiB de memorie RAM. Programele pot fi recuperate dintr-o memorie flash externă de până la 16 MiB prin intermediul QSPI. Partea radio are o interfață IEEE 802.11b Wi-Fi b/g/n, cu un comutator T/R și o antenă integrată "*balun*", LNA (Low Noise Amplifier) pentru recepție, respectiv un amplificator de putere și adaptor de impedanță pentru transmisie [135].

Ceea ce a fost decizia pe care am luat-o pentru alegerea ESP8266 este un convertor analog digital (ADC) integrat de 10 biți la care se pot conecta diverși senzori. Pentru demonstratorul de tele-măsurare în Cloud am folosit un senzor de temperatură LM35 (Texas Instruments).

Pentru a păstra avantajul miniaturizării, seturile de configurații și conexiuni recomandate de producătorul ESP8266 sunt grupate într-o serie de module miniaturizate – am ales modulul 12 care are acces direct la convertorul analogic-digital capabil să achiziționeze datele de la senzori (figura 29).



Figura 29: Exemplu de modul tip 12 de la AI THINKER (se observă antena pe cablaj imprimat)

Modulul este montat pe o mică placă de dezvoltare Wemos D1 (figura 30) cu firmware scris în LUA ("Luna" – în portugheză – acronimul unui limbaj simplificat de programare a sistemelor "embedded", cu micro-calculator încorporat). Resurse pentru programarea Wemos D1 sunt disponibile în platforma open-source IoT NodeMCU, integrată cu mediul Arduino [136].

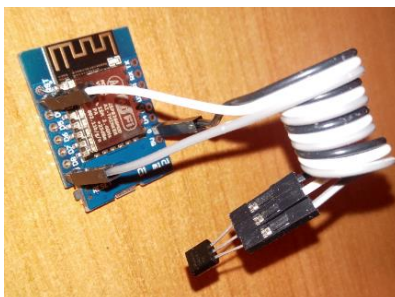


Figura 30: Conectarea senzorului de temperatură LM35 (Texas Instruments) - ADC/alimentare/GND

5.1.2. Serverul de instrumentație Cloud și tele-măsurarea bazată pe TCP

Infrastructura Cloud se bazează pe hypervisorul Citrix XenServer care rulează pe un server "multi-blade" cu sisteme Dell Power Edge 1950. Accesul la distanță e reglementat de soluția SoftEther VPN, iar Intranetul se bazează pe un router Gigabit de la Cisco. Serverul de instrumentație – bazat pe soluții National Instruments (NI) centrate pe LabVIEW – rulează pe o gazdă virtuală (host XenServer). Pentru a permite accesul public la serverul de instrumentație, porturile necesare ("incoming" și "out-going") sunt redirecționate de la public host la mașina virtuală Windows ce rulează LabVIEW [137].

În figura 31 este prezentată colectarea TCP a datelor în Cloud, de la senzorii aflați la distanță (dispozitive IoT bazate pe ESP8266, cum am arătat anterior), implementată pe mașina virtuală Windows menționată – ca instrument virtual LabVIEW (VI – Virtual Instrument).

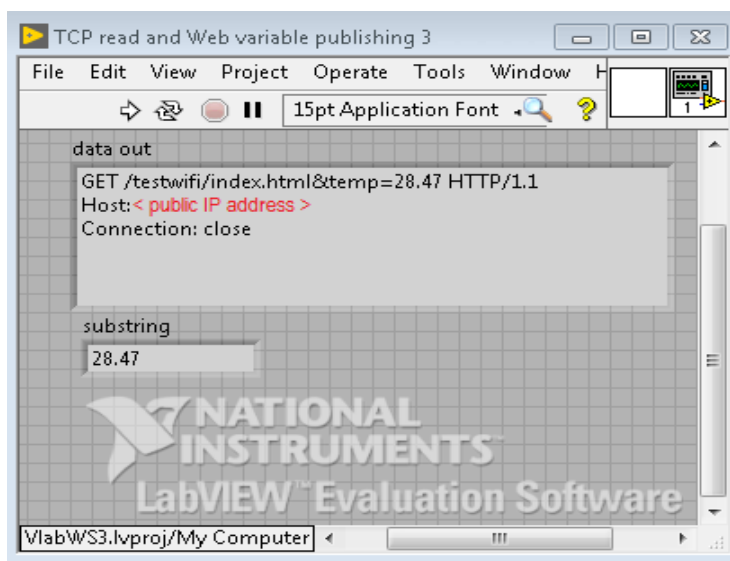


Figura 31: Colectarea datelor de la senzorii aflați la distanță – Panoul TCP VI

Serviciile web pe care le-am implementat au un avantaj important: oferă date relevante "în timp real" (constantele de timp ale tele-transmisiei fiind cu unul-două ordine de mărime sub constantele de timp ale dispozitivelor IoT), fără a transmite informații excesive suplimentare (de exemplu informații despre procedura de achiziție a datelor, etc.)

5.1.3. Caracteristicile cheie ale soluției în contextul Big Data

Este la modă "*transformarea digitală*" ("Digital Transformation") care se extinde, treptat, de la domeniul economic spre întreaga noastră activitate. Trăind într-o lume în care riscăm să fim inundați de tot mai multe date, una din provocările "transformării digitale" este tocmai gestiunea "masivelor de date" ("Big Data", "Data Mines" etc), cu structură interogabilă sau nu (non- SQL) a bazelor de date. Multe dintre aceste date provin de la dispozitivele IoT (senzori și traductoare inteligente) – implicit, multe problemelor sunt legate de conectivitatea senzorilor la sistemele superioare ierarhic, nu numai pentru transferul datelor achiziționate dar și pentru a fi ușor de gestionat în sensul scalabilității și al securizării [142].

Serviciile web NI reprezintă partea critică a sistemului, deoarece aici se formează datele pentru alimentarea sistemului cu informația de procesat. Această componentă este responsabilă pentru conectarea cu senzorii și publicarea datelor pe care le consumă ca variabile partajate (Shared Variables) prin intermediul serviciilor REST de timp real (în raport cu constantele de timp ale proceselor industriale, domotice sau de mediu monitorizate).

5.1.4. Panoul de control bazat și publicat pe web

Aplicația dezvoltată, numită: *Control Board*, este un „Thin Client”, adaptabil la terminal, reprezentat și dezvoltat nu ca o aplicație instalată (care rulează local), ci ca o aplicație bazată pe web (pe informația publicată de serverul Cloud de instrumentație), disponibilă și compatibilă cu orice navigator (browser) web popular. "Control Board" este dezvoltat pentru conectarea și gestionarea centralizată a serviciilor de telematică și constă dintr-o interfață grafică cu utilizatorul (GUI) care poate fi lansată prin accesarea unei adrese web fără a fi nevoie să se instaleze module suplimentare și plugin-uri terțe, care pot genera încălcări ale securității, pot crea un consum necesar de resurse și pot provoca dificultăți și complicații pentru utilizator din cauza dependențelor suplimentare.

5.1.5. Implementarea serviciilor web de prezentare

Primul ecran al „Control Board” este proiectat să se conecteze la serverul Cloud de instrumentație (pe care rulează toate sub-sistemele software intensive computațional). După stabilirea conexiunii, „Thin Client”-ul a fost programat să ofere o listă a serviciilor web publicate pe serverul Cloud. Serviciile NI invocate fac parte din setul de administrare a sistemului LabVIEW ("LVWSSysAdmin"):

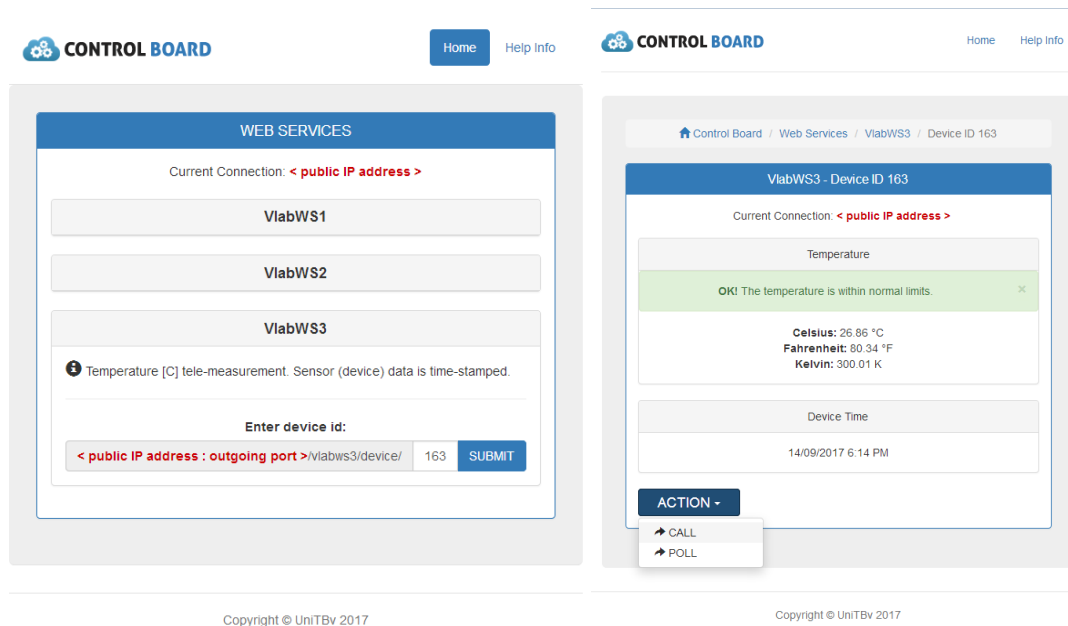


Figura 32: Afișarea unei scurte descrieri a serviciilor web publicate (stânga) și a opțiunii de prezentare - un exemplu de acțiuni, unități etc. (dreapta)

În imaginea următoare este prezentată afișarea datelor de tele-măsurare de la dispozitivul IoT (cu ESP8266, senzor LM35 și legătură prin Wi-Fi Access Point) prin „Control board”, pe două terminale inteligente diferite, cu sistemele de operare Windows Mobile și Android (fără instalare de plug-ins suplimentare, accesarea se face prin intermediul navigatorului web nativ fiecărui sistem de operare).

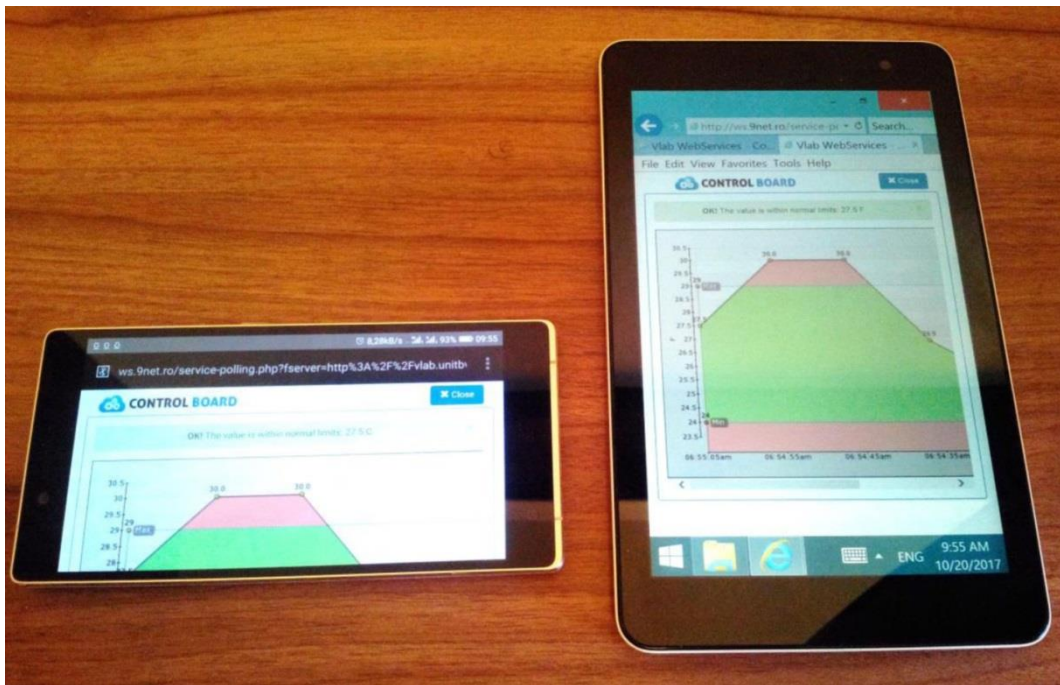


Figura 33: Afișarea panoului de control pe două dispozitive diferite, cu sistemele de operare Windows Mobile și Android și cu navigatoare diverse

5.2. Contribuție la accesul receptiv la terminal în IoT

5.2.1. Concept

Plasticitatea este capacitatea unei interfețe cu utilizatorul de a suporta variațiile – atât ale caracteristicilor fizice ale sistemului, cât și ale mediului – păstrând în același timp utilitatea. Accesul receptiv și adaptarea conținutului la terminal sunt caracteristici principale ale unei interfețe de utilizator plastice [146].

Așa cum am arătat, web-ul modern și Cloud Computing sunt compatibile cu o gamă largă de terminale mobile (smartphones sau tablete) care rulează "clienți subțiri" ("Thin Clients") [147]. Acești clienți sunt compatibili cu dispozitivele portabile, "cross-platform & cross-device", pot centraliza sarcinile de informare și gestionare, sunt mai puțin costisitori și - ceea ce îi definește ca "thin" - sunt proiectați să consume mai puțină energie și resurse pentru a-și îndeplini sarcinile.

5.2.2. Reactivitate și receptivitate

„Thin client”-ul pe care l-am dezvoltat are un design receptiv (la cerințele diferite ale terminalelor) – “responsive design” (utilizând standardizate HTML5, CSS3, JavaScript și jQuery) [150]. Aceasta este o caracteristică esențială, astăzi mai mult ca niciodată, din cauza noilor tipuri de dispozitive dezvoltate continuu.

Logo-urile sunt reduse ca imagini sau sunt înlocuite cu icon-uri mai mici pentru grafica detaliată [154]. Bara de meniu verticală care are nevoie de multă lățime este repositionată ca o listă restrânsă care poate fi vizualizată numai la clic, așa cum este prezentat în figura de mai jos:

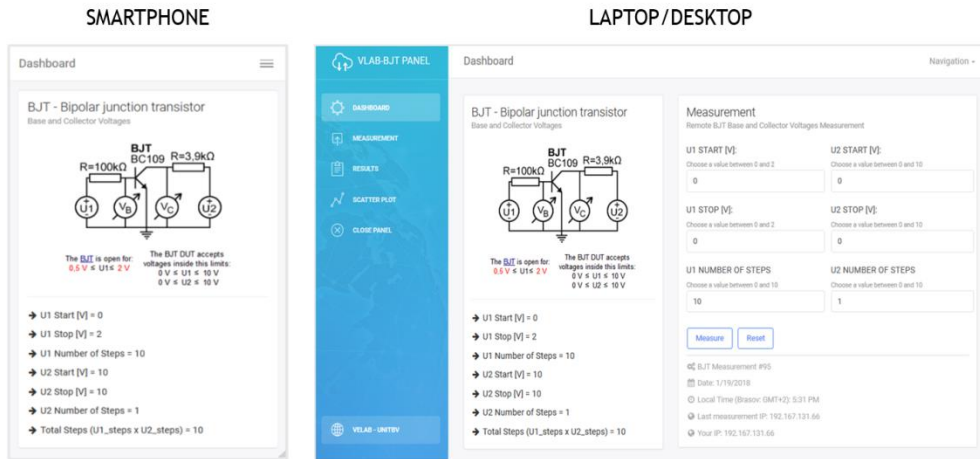


Figura 34: “Thin client” (“clientul subțire”) afișat pentru o rezoluție de dispozitiv tip smartphone (stânga) și tabletă (dreapta)

Datele măsurate sunt colectate și analizate utilizând serviciile web și afișate în paralel, ca rezultate măsurate și simulate:

Experimental Results				Simulated Results			
U1 (DATA IN)	V_C (DATA OUT)	V_B (DATA OUT)	U2 (DATA IN)	U1 (DATA IN)	V_C (DATA OUT)	V_B (DATA OUT)	U2 (DATA IN)
EXPERIMENTAL				SIMULATED			
0.00	10.00	0.035	10.00	0.00	10.00	0.001	10.00
0.20	10.00	0.235	10.00	0.20	10.00	0.200	10.00
0.40	10.00	0.435	10.00	0.40	10.00	0.400	10.00
0.60	9.39	0.586	10.00	0.60	9.77	0.585	10.00
0.80	6.57	0.630	10.00	0.80	7.10	0.652	10.00
1.00	3.36	0.650	10.00	1.00	3.72	0.674	10.00
1.20	0.65	0.660	10.00	1.20	0.68	0.687	10.00
1.40	0.20	0.665	10.00	1.40	0.24	0.689	10.00
1.60	0.17	0.665	10.00	1.60	0.21	0.689	10.00
1.80	0.16	0.665	10.00	1.80	0.19	0.689	10.00
				2.00	0.18	0.689	10.00

Figura 35: Rezultatele experimentale (stânga) și simulate (dreapta)

5.2.3. Panoul "CON-INTEL Dashboard"

"CON-INTEL Dashboard" este un "Thin Client" pe care l-am dezvoltat în anul 2018 în cadrul Proiectului de cercetare PN-III-P2-2.1-PTE-2016-0064 "CON-INTEL – Măsurarea consolidată și transmiterea parametrilor energetici spre punctele de colectare", finanțat de UEFISCDI (a se vedea paragraful 7.3.5). "CON-INTEL Dashboard" este construit pentru citirea de la distanță, prin Internet, a consumului de energie electrică activă și reactivă, prin intermediul aceluiași gen de server de instrumentație descris în paragraful 5.2.1 menit să centralizeze tele-măsurări din Smart-Grid, de la "contoare Cloud". Panoul cuprinde o interfață grafică cu utilizatorul (GUI), care poate fi conectată și angajată prin simpla accesarea a unei adrese web (URL) fără necesitatea instalării de module suplimentare și "plugins" de la terți (care pot genera vulnerabilități de securitate cibernetică, pot genera consumuri inutile de resurse și pot complica experiența utilizatorului din cauza dependențelor suplimentare) [158].

Deoarece în ziua de azi o aplicație poate fi accesată dintr-o varietate de dispozitive cu o mare diversitate de rezoluții, "Thin Client"-ul a fost dezvoltat utilizând Bootstrap Framework, care acționează ca un motor web, pentru a gestiona și manipula conceptul de abordare de proiectare receptivă și responsivă (astfel încât se oferă suport pentru o gamă largă de terminale pentru o singură aplicație și se asigură navigarea optimizată și calitatea experienței (QoE)) [159].

În particular, elementele de bloc sunt tratate diferit. În secțiunile în care grila de proiectare utilizează mai multe coloane, blocurile se adaptează variabil la lățimea ferestrei de vizualizare respective [160]. Pentru rezoluții mici, dacă o reprezentare nu se mai potrivește una lângă cealaltă, este poziționată alternativ (figurila 36).

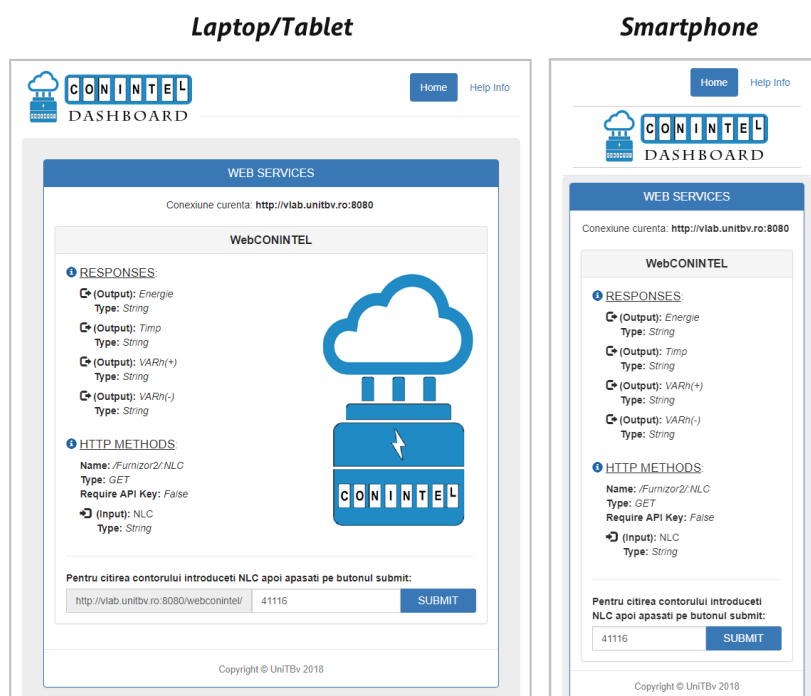


Figura 36: Poziționarea alternativă și scalarea la un maxim activat de elementele respective Laptop/Tablet (stânga) & Smartphone (dreapta)

5.3. Sumarul capitolului

Cercetarea s-a axat pe serviciile web pentru telematică și prezentarea îmbunătățită a datelor colectate de Cloud, bazată pe soluția specială de servicii web LabVIEW, achiziția de date poate declanșa furnizarea atât a conținutului HTML static, cât și a metodelor JS (JavaScript). În acest fel, transmiterea datelor poate include informații despre prelucrarea ulterioară a acestora.

Această caracteristică importantă are avantajul de a permite autorizarea paginilor HTML5 care prezintă datele senzorilor de tele-măsurare într-un mod ușor de utilizat.

Interfața grafică dinamică (GUI) a centrului de control („Control Board”) de tip "Thin Client" poate fi dezvoltată în continuare pentru a accepta nu doar comenzile principale ("Call", "Poll" etc.) pentru a obține valori actualizate (de măsurare sau stare) dar și aspectul și aranjarea indicatorilor/comenzilor cat și/sau a unităților (de exemplu, Fahrenheit în loc de Celsius etc.)

Datele brute "tele-măsurate" pot fi preluate împreună cu metodele proprii de procesare însoțitoare. Acestea pot fi completate și cu "metadate" cu privire la semnificație sau reprezentare. Astfel de metadate pot fi modificate într-un singur loc (de ex., pe server, fiind actualizate automat pentru toți clienții). Pentru a evita ca metodele să devină o „cheltuială” pentru valorile tele-măsurate, ar putea fi asigurat "caching"-ul, inclusiv în antetul solicitării GET, un fel de marcare temporală "neschimbată din..." (*not modified since*). Prin intermediul *serviciilor REST*, se face o nouă abordare a *controlabilității* (*tranzițiile în automatul controlat prin stare al modelului comportamental IoT* sunt comandate prin cereri-metode HTTP), expunând capabilități de tip „controale” - *„controlul ca serviciu”*.

Deoarece este dezvoltat cu ajutorul unei arhitecturi de tip microservicii, produsul software de tip „Thin Client” este mai ușor de întreținut, corectat și actualizat, conducând la capacități mai agile pentru a răspunde schimbărilor de mediu de astăzi. Separarea componentelor unei aplicații face mult mai puțin probabil ca un bug sau o problemă să afecteze întregul sistem, iar orice "defecte" pot fi izolate individual, reparate și repornite fără a întrerupe funcționalitatea întregii aplicații.

Una dintre abordările cele mai fezabile pentru a asigura accesul la serviciile de laborator virtual printr-o rețea, precum și pentru a oferi o interfață prietenoasă și intuitivă pentru utilizatori este prin utilizarea "clienților subțiri" adaptivi și receptivi la terminal. Soluția dezvoltată operează în mod consecvent, cu dependențe minime și cu resurse computaționale aproape ne semnificative. Aceste caracteristici permit o gestionare foarte simplă a controlului centralizat sau descentralizat, cu reducerea cheltuielilor generale și un număr aproape nelimitat de „clienți subțiri”. Acest lucru este în concordanță cu paradigma echilibrării Edge Computing/Cloud Computing și cu tendința de creștere a capacităților viitoarelor smartphone-uri (cele moderne încorporează mai multă capacitate de calcul, capacitate de stocare și de comunicare decât clădirile multi-computer de tip "centru de date" din secolul trecut). Numărul nelimitat de „clienți subțiri” este un aspect critic în furnizarea de servicii de telematică, deoarece probabilitatea de a avea o densitate mare de clienți este semnificativă, în special în mediile universitare.

Implementarea aplicației descrisă oferă posibilitatea de a instanția și a declanșa măsurători la distanță, precum și o vizualizare bazată pe servicii de web a rezultatelor măsurătorilor telemetrice și poate adapta prezentările sale pe baza caracteristicilor terminalelor într-un mod clar și ușor de utilizat și vizualizat. La efectuarea măsurătorilor, rezultatele (de obicei valori numerice) sunt cele mai importante, totuși interpretările grafice diferite au o contribuție importantă la o mai bună înțelegere a rezultatelor obținute. Acest grad mai ridicat de vizualizare, combinat cu caracteristica de design receptiv, are, din perspectiva utilizatorului, un impact crucial asupra calității experienței (QoE).

6. Dezvoltare Web și managementul semantic pentru IoT

6.1. Semantică și ontologii

6.1.1. Web-ul Semantic

Web-ul Semantic este o extensie a World Wide Web prin intermediul standardelor elaborate de către World Wide Web Consortium (W3C) [161]. Cu scopul de a muta accentul în manipularea informației de la sintaxă (mod de scriere etc – și erorile tipice) la *semnificație* (sensul, înțelesul informației, exprimat de metadate – etichete semantice asupra categorisirii, utilizării frecvente și altele care țin de *rostul* informației), aceste standarde promovează formate comune de date și protocoale de schimb pe Web, printre cele mai populare fiind Resource Description Framework (RDF), RDFa (Resource Description Framework in Attributes), Microformats și Microdata [162]. Web-ul Semantic oferă astfel un cadru comun (orientat mai degrabă pe informație și nu pe conexiune), care permite partajarea și reutilizarea datelor între parteneri – de la aplicații partenere și până la parteneri umani, instituționali sau comunitari. Prin urmare, Web-ul Semantic este privit ca un integrator pentru diferite conținuturi, aplicații și sisteme de informare.

Micro-formatele extind sintaxa HTML pentru a crea marcaj semantic – care poate fi citit de dispozitive – despre obiecte, persoane, organizații, evenimente sau produse.

6.1.2. Soluții Web semantice

Web-ul semantic implică publicarea în limbaje special concepute pentru descrierea datelor: RDF, OWL și XML [164].

Exemplu de marcare HTML utilizat într-o pagină web non-semantică:

```
<item>continut</item>
```

Exemplu de marcare a unor informații similare într-un document Web semantic:

```
<item rdf:about="http://exemplu.ro/web-semantic/">Continut semantic</item>
```

Rețeaua orientată conținutul semantic este numită de către Tim Berners-Lee ca și GGG - Giant Global Graph spre deosebire de WWW (World Wide Web) care este bazat doar pe HTML, guvernate de principii simple ale democratizării *instruirii* [166]:

1. o adresă URL trebuie să indice datele;
2. oricine accesează adresa URL trebuie să primească înapoi date, iar relațiile din date trebuie să indice URL-uri suplimentare cu date.

6.1.3. Provocări ale Web-ului Semantic

Web-ul Semantic include mai multe provocări pentru viitoarele soluții de raționament automat în ceea ce privește gestiunea conținutului semantic [167]:

- Vastitatea;
- Neclaritatea;
- Incertitudinea;
- Inconsecvența;
- Înșelăciunea;

Această listă de provocări este mai degrabă ilustrativă decât exhaustivă și se concentrează asupra provocărilor privind "logica unificatoare" (a interogărilor, a reglementărilor și descrierilor ontologice), guvernată de gestiunea "dovezilor" și de managementul "încrederii" în Webul Semantic.

6.2. Standarde și taxonomii

Așa cum am arătat în paragraful 2.4.2, în contextul lucrării de față, *ontologia* e un ansamblu de nume și reprezentări menite să dea sens cunoștințelor prin structurarea categoriilor, proprietăților și relațiilor dintre conceptele unui domeniu.

6.2.1. Resource Description Framework

Cadrul pentru descrierea resurselor – RDF (Resource Description Framework) este o specificație W3C (World Wide Web Consortium) concepută ca un model pentru *metadate*. RDF s-a dezvoltat apoi și ca metodă generală pentru descrierea conceptuală sau modelarea informațiilor care sunt implementate în resursele web, utilizând o varietate de notații sintactice și formate de serializare a datelor [169]; este, de asemenea, utilizat în aplicațiile de gestionare și management al cunoștințelor.

RDF este un model abstract cu mai multe formate de serializare a datelor (de exemplu pentru înregistrare în fișiere), astfel încât *codificarea specifică pentru resurse* reprezentată de tripletele *subiect–predicat–obiect* variază de la un format la altul. Acest mecanism de descriere a resurselor este o componentă majoră a activității Web-ului Semantic ca etapă evolutivă a World Wide Web în care software-ul automatizat poate stoca, schimba și utiliza informația distribuită pe web și citită de mașină, permițând utilizatorilor să se ocupe de informații cu o mai mare eficiență și siguranță. Modelul simplu de date al RDF și abilitatea de a modela concepte disparate și abstracte au condus, de asemenea, la utilizarea sa în creșterea în aplicațiile de management al cunoașterii ("knowledge management").

6.2.2. RDFa (Resource Description Framework in Attributes)

RDFa (Resource Description Framework in Attributes) este o recomandare W3C care adaugă un set de extensii – la nivel de attribute HTML, XHTML și diferite tipuri de documente bazate pe XML – pentru încorporarea metadatelor în documentele Web.

Maparea modelului de date RDF permite utilizarea sa pentru încorporarea expresiilor RDF de tip *subiect-predicat-obiect* în documentele XHTML [171]. De asemenea, permite extragerea modelului triplei RDF de către agenți software compatibili ("compliant user agents").

Esența RDFa este de a furniza un set de atribute care pot fi folosite pentru a transporta metadatele într-un limbaj de tip XML. În plus, RDFa poate beneficia de *accesibilitatea web*, deoarece mai multe informații sunt disponibile pentru *tehnologia asistivă* (-de asistență, "assistive technology").

6.2.3. Microformat

Un microformat (uneori abreviat μ F) este o abordare recomandată de W3C pentru marcarea semantică și care utilizează etichete HTML/XHTML pentru a transmite metadate suplimentare și alte atribute în paginile web și în alte contexte care acceptă (X)HTML, cum ar fi RSS (a se vedea paragraful 6.2.1) [172]. Această abordare permite software-ului să proceseze automat informațiile destinate utilizatorilor finali (cum ar fi informațiile de contact, coordonatele geografice, evenimentele din calendar și alte informații similare).

Microformatele pot elimina acest decalaj de *lizibilitate* dintre „*human-readability*” și „*machine-readability*” prin atașarea reprezentării semantice, evitând astfel alte metode mai complexe de prelucrare automată, cum ar fi prelucrarea limbajului natural (NLP - „*natural language processing*”) sau „*screen scraping*” (drenarea - extragerea informației din ecranele afișate) [173]. Utilizarea, adoptarea și procesarea microformatelor permite indexarea, căutarea, salvarea sau încrucișarea datelor (mixarea formatelor), astfel încât informațiile să poată fi refolosite sau combinate.

Mai multe extensii de browser, cum ar fi *Operator* pentru *Firefox* și *Oomph* pentru *Internet Explorer*, oferă posibilitatea de a detecta microformatele dintr-un document HTML.

6.2.4. Microdata

Microdata este o specificație de tip extensie HTML folosită pentru a adăuga metadatele în cadrul conținutului paginilor web. Motoarele de căutare, crawlerele web (menționate la începutul paragrafului 6.2) și browserele pot extrage și procesa *microdatele* de pe o pagină web și le pot folosi pentru a oferi utilizatorilor o experiență de navigare mai bogată. Motoarele de căutare beneficiază în mare măsură de accesul direct la aceste date structurate ceea ce le permite să interpreteze informațiile de pe paginile web și să ofere utilizatorilor mai multe rezultate relevante [174]. Microdata utilizează un vocabular de tip perechi *nume-valoare* pentru a atribui valori proprietăților ce descriu un element. Microdata este o încercare de a oferi o metodă mai simplă de adnotare a elementelor HTML (cu etichete care pot fi citite de mașină) decât abordările similare - RDFa și microformatele.

6.2.5. JSON-LD (JavaScript Object Notation for Linked Data)

JSON-LD (JavaScript Object Notation for Linked Data), este o recomandare a Consorțiului World Wide Web reprezentând o metodă de codare a datelor asociate utilizând JSON. Scopul este acela de a solicita cât mai puțin efort dezvoltatorilor în a-și transforma documentele JSON existente în JSON-LD [176]. Acest lucru permite ca datele să fie serializate într-un mod similar cu cel al JSON tradițional.

JSON-LD este creat în jurul conceptului de "context" pentru a furniza mapări suplimentare de la JSON la un model RDF. Contextul leagă proprietățile obiectului într-un document JSON cu conceptele dintr-o ontologie.

Pentru a mapa sintaxa JSON-LD la RDF, JSON-LD permite ca valorile să fie forțate la un tip specificat sau să fie etichetate cu un limbaj. Un context poate fi încorporat direct într-un document JSON-LD sau introdus într-un fișier separat și referit din documente diferite (de la documente tradiționale JSON printr-un antet HTTP Link)

Codificarea JSON-LD este utilizată de Schema.org și Google Knowledge Graph și este folosită în principal pentru activitățile de optimizare a motoarelor de căutare (SEO – Search Engine Optimization). De asemenea, JSON-LD este utilizat pentru aplicații cum ar fi informatica biomedicală – ce includ informații de *proveniență*.

6.3. Web Semantic al obiectelor

În prezent lumea dezvoltată și tehnologizată este plină de senzori. Cu toate acestea, senzorii sunt în mod obișnuit blocați în sisteme închise, *uni-modale*. Pentru a-și elibera întregul potențial, accesul la senzori ar trebui deschis, astfel încât datele și serviciile din sistem să poată fi integrate cu datele și serviciile disponibile în alte sisteme informatice, facilitând aplicații noi și servicii controlate în timp real de evenimente din lumea reală.

Arhitectura unui "*Web Semantic al obiectelor*" (SWoT - Semantic Web of Things) reprezintă o infrastructură de servicii care face implementarea și utilizarea *aplicațiilor semantice care implică senzori conectați la Internet* aproape la fel de ușoară ca și construirea, căutarea și citirea unei pagini web [177]. Conectivitatea la Internet nu necesită doar integrare la nivel de rețea (IP), ci și *integrare la nivel de aplicație* pentru a permite accesul structurat la datele senzorilor. Pentru a permite raționamentul automat bazat pe datele de la senzori (de exemplu, găsirea de locuri de parcare gratuite aproape de o anumită sală de ședință), acești senzori, ieșirea lor și interfața lor cu lumea reală trebuie să fie descrise într-un format care poate fi citit de computer compatibil cu formatele de date existente pe web.

6.3.1. Rețele semantice de senzori

Semantic Sensor Web (SSW) este o sinergie a tehnologiilor senzorilor și a Web-ului semantic. Codarea descrierilor senzorilor și a datelor de măsurare de la senzori cu limbajele Web semantice permite o reprezentare mai expresivă, un acces îmbunătățit și o analiză formală a resurselor senzorilor. SSW adnotează datele senzorilor cu metadate semantice – nu numai spațiale și temporale dar și tematice [179].

Această tehnică se bazează pe eforturile actuale de standardizare din cadrul Open Geospatial Consortium (OGC) - Sensor Web Enablement (SWE) și le extinde cu tehnologiile Web Semantic pentru a oferi descrieri îmbunătățite și acces extins la datele senzorilor.

6.3.2. Modelarea și adnotarea semantică a datelor senzorilor

Ontologiile și alte tehnologii semantice pot fi tehnologii de bază pentru rețelele de senzori, deoarece ele pot ameliora *interoperabilitatea și integrarea*, pot facilita raționamentul automat (bazat pe interpretarea multitudinii de date colectate), clasificarea și alte tipuri de proceduri care nu sunt încă incluse în recomandările OGC [180].

SSW va permite rețelei, senzorilor și datelor rezultate să fie organizate, instalate și gestionate, interogate, înțelese și controlate prin *specificații la nivel înalt*. Ontologiile pentru senzori oferă un cadru pentru descrierea acestora. Aceste ontologii permit clasificarea și specificarea capacităților senzorilor, a măsurătorilor efectuate (tip, format, precizie etc), a provenienței măsurătorilor și permit raționamentul (diagnoza, decizii de control etc) atât asupra senzorilor individuali, cât și asupra unui sub-sistem de senzori interconectați,

gestionați ca "macro-instrument". Ontologiile senzorilor conforme recomandărilor OGC au avantajul de a permite o "cartografiere a modelelor" astfel încât adnotarea semantică a descrierilor senzorilor și a serviciilor care susțin schimbul de date al de la senzori precum și gestionarea rețelelor de senzori urmează să fie *integrată cu adnotarea semantică a serviciilor web*. Aceste cercetări se realizează sub egida W3C Semantic Sensor Network Incubator Group (SSN-XG) [181].

6.3.3. Rețelele senzorilor semantici W3C

Consortiul World Wide Web (W3C) a inițiat SSN-XG pentru a dezvolta ontologia Semantic Sensor Network (SSN) destinată modelării dispozitivelor, sistemelor, proceselor și măsurărilor senzorilor [182].

"Senzorul Web" – în sensul IoT al lucrării de față - este un dispozitiv special al infrastructurii informatice – bazată pe Web – pentru colectarea, modelarea, stocarea, preluarea, partajarea, manipularea, analizarea și vizualizarea informațiilor de la și despre senzori și traductoare. Lipsa standardizării în domeniu a dus la lansarea recentă, de către OGC, inițiativei SWE (Sensor Web Enablement) pentru a dezvolta o suită de specificații referitoare la modelele de date de la senzori și la serviciile bazate pe senzori web. Senzorii care codifică fenomenele observate sunt prin natura lor "opaci" (adesea în formate binare proprietare); prin urmare, metadatele joacă un rol esențial în gestionarea datelor senzorilor (care devin astfel "transparente" – independente de nivelele inferioare OSI "mo-dem", "co-dec" etc). O rețea de senzori "bogată semantic" ar putea furniza informații spațiale, temporale și tematice esențiale pentru descoperirea și analizarea datelor.

Metadatele spațiale furnizează informații privind localizarea senzorilor, fie în ceea ce privește un sistem de referință geografic, o referință locală sau o locație reprezentată prin numele ei. Referința locală este utilă în special atunci când un senzor este atașat unui obiect în mișcare, cum ar fi o mașină sau un avion. Deși locația senzorului se schimbă în mod constant, locația acestuia poate fi determinată static față de obiectul în mișcare. În plus, datele transmise de senzori la distanță (de exemplu, foto- video de la sateliți), necesită modele spațiale complexe pentru a reprezenta domeniul monitorizat, care poate fi mult diferit de locația senzorului.

6.4. Contribuție la accesibilitatea și managementul interfețelor semantice în IoT

Internet / Intranet sunt în continuă creștere, aplicațiile web devin din ce în ce mai complexe, iar web-ul semantic și-a pus deja amprenta pe evoluția arhitecturilor de rețea moderne. Dezvoltatorii și designerii web folosesc adesea comenzi noi și personalizate integrate în aplicații care nu pot fi reprezentate doar utilizând instrumentele tradiționale de limbaj de marcare. Acestea includ meniuri "drop-down", panouri ("tabs"), structuri ierarhice, "glisoare", câmpuri care permit o intrare oferind în același timp dinamic sugestii de introducere și asistență sintactică – așa cum am arătat în capitolul 5, toate împreună construiesc conceptul de *Rich Internet Applications* [183].

Accesibilitatea este caracteristica unui dispozitiv, a unui serviciu, a unei resurse sau a unui mediu care poate fi accesat cu ușurință de orice tip de utilizator. Termenul este asociat în mod obișnuit cu posibilitatea ca și persoanele cu capacitate senzorială, motrică sau psihică redusă (afectată de invaliditate temporară sau permanentă) să acceseze și să se deplaseze independent în medii fizice și informatice, să acceseze și să utilizeze în mod autonom

conținutul cultural sau să beneficieze de sistemele și resursele IT disponibile în mod obișnuit – prin utilizarea tehnologiilor de asistență sau prin respectarea cerințelor speciale de accesibilitate a produselor [184].

În acest context, soluțiile de accesibilitate sunt dezvoltate pentru a favoriza reducerea sau eliminarea decalajului dintre cei care pot accesa în mod independent resursele web și cei care nu pot face aceasta (în special persoanele cu deficiențe de vedere).

6.4.1. Aplicații Rich Internet Accesibile

Tehnologiile emergente au democratizat accesul la Internet. Dar există și un număr de persoane cu dizabilități care nu pot beneficia încă de aceste noi tehnologii. Tipul de conținut precum rolul, starea și proprietățile conținutului unei aplicații, care sunt actualizate în timp real, sunt adesea indisponibile beneficiarilor tehnologiilor de asistență [185].

Pentru a obține accesibilitatea și pentru utilizatorii cu deficiențe de vedere, am extins limbajul de markup HTML5 cu metadatele semantice WAI-ARIA așa cum se arată în figura 37.

```
1 <div class="tabs">
2   <div role="tablist"
3     aria-label="Research">
4     <button
5       role="tab" aria-selected="true" aria-controls="wai-aria-tab" id="wai-aria">
6       Accessible Rich Internet Application
7     </button>
8     <button
9       role="tab" aria-selected="false" aria-controls="amp-tab" id="amp" tabindex="-1">
10      Accelerated Mobile Pages
11    </button>
12    <button
13      role="tab" aria-selected="false" aria-controls="semantic-web-tab" id="semantic-web"
14      tabindex="-1" data-deletable="">
15      Semantic Web & Web Accessibility
16    </button>
17  </div>
```

Figura 37: Exemplu metadate semantice WAI-ARIA care extind marcajul HTML5 pentru butoanele de control

WAI-ARIA este o specificație care oferă ajutor pentru a descrie caracteristicile secțiunilor de conținut auto-dezvoltate, făcându-le identificabile și utilizabile de către utilizatorii tehnologiilor de asistență [186]. Etichetele pot fi ușor procesate de un program care cunoaște aceste convenții, cum ar fi un cititor de ecran ("text to speech screen reader") sau un afișaj Braille. Un cititor de ecran, numit și o aplicație „citită cu voce tare” ("text to speech"), este un software care oferă persoanelor cu deficiențe de vedere o interfață de utilizator alternativă în locul modului text sau al unei interfețe grafice de utilizator (GUI). Un cititor de ecran comunică informațiile care sunt de obicei afișate pe ecran utilizând dispozitive de ieșire non-vizuale.

Comenzile și textele sunt reproduse acustic în majoritatea cazurilor prin intermediul unei plăci de sunet (prin intermediul sintetizatoarelor de vorbire) sau tactile prin intermediul unui afișaj Braille. Figura 38 prezintă vizualizarea frontală a elementelor blocului HTML5, extinsă cu marcarea WAI-ARIA, care este afișată în browserul web utilizând plugin-ul ChromeVox Screen Reader, atunci când un utilizator interacționează și vizualizează aplicația pentru a accesa informațiile transformând valorile sale în text; în acest fel utilizatorii pot găsi toate caracteristicile pe care aplicațiile le pun la dispoziție.

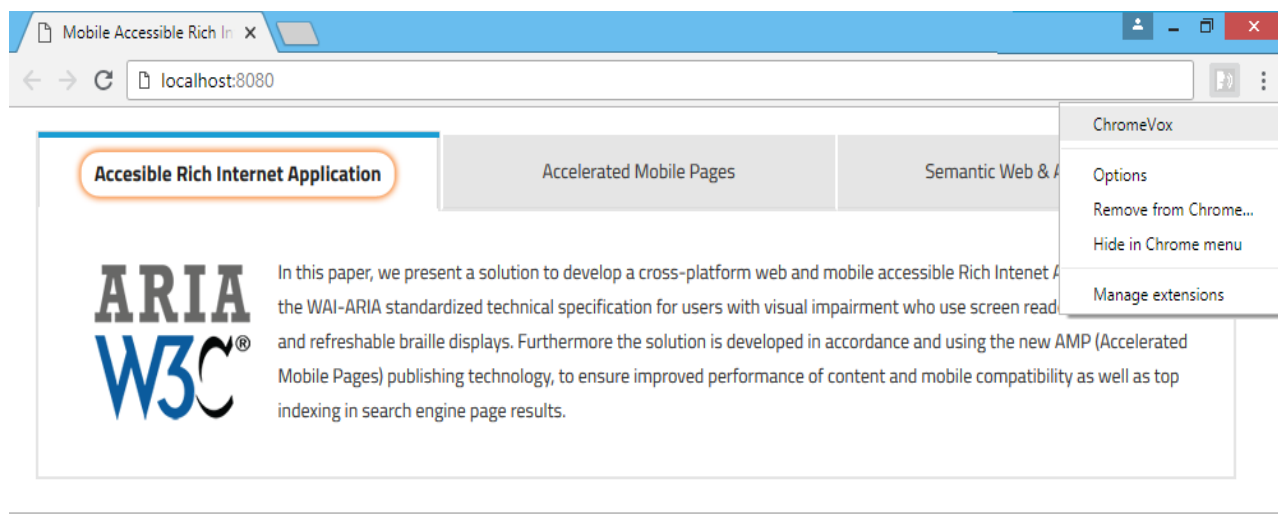


Figura 38: Vizualizare frontală afișată în browser utilizând ChromeVox Screen Reader

6.4.2. AMP - Accelerated Mobile Pages

Accelerated Mobile Pages (AMP) este un cadru ("framework") cu sursă deschisă ("open-source"), "cross-platform", care poate crește semnificativ viteza de încărcare a site-urilor mobile. AMP se bazează pe reducerea ponderii CSS și JavaScript, pe o rețea de difuzare a conținutului și pe cod HTML personalizat. Proiectul Accelerated Mobile Pages este susținut de Google și în comparație cu aplicațiile optimizate pentru dispozitive mobile sau cu aplicațiile receptive ("responsive" – a se vedea capitolul 5), documentele AMP sunt încărcate mult mai rapid chiar și peste conexiunile cu lățime de bandă redusă și sunt redactate mai repede în browserele web populare bazate pe HTML [187]. În contextul acestui paragraf, accesibilitatea înseamnă „lipsa obstacolelor”. Soluția dezvoltată (optimizată pentru dispozitive web/mobile și utilizarea tehnologiilor standardizate și a limbajelor de marcare valide) vizează în primul rând accesibilitatea la computer - atât pentru persoanele cu limitări fizice, tehnice sau legate de vârstă (deficiențe vizuale), cât și pentru crawler-ele web, cu care motoarele de căutare captează conținutul unei pagini (pentru a evalua conținutul din punct de vedere semantic, pentru a genera rezultate mai bune ale căutărilor viitoare), corelând accesibilitatea web pentru oameni și accesibilitatea pentru roboți (și alte tipuri de "bots") [188].

6.5. Contribuție la managementul resurselor utilizând capabilități offline progresive și inter-platformă

Așa cum am arătat în capitolul 5, *Thin Clients* sunt necesari atunci când clienții obișnuiți *fat clients* sunt prea costisitori, dificil de optimizat sau pentru că au nevoie de mai multă putere sau energie de calcul decât sunt disponibile de la terminalele "low-end" (de exemplu, tablete sau smartphones).

Un Thin Client bazat pe web este un program de aplicație care funcționează în conformitate cu modelul client-server, unde tot software-ul rulează pe server și numai prezentarea este livrată pe dispozitiv (de exemplu, aplicații GIS sau de tele-măsurare unde Thin Client este, în principiu, browserul web) [189]. În web-ul modern, orientat către portabilitate, Cloud computing și virtualizare, tendința este de a se dezvolta Thin Clients cat mai optimizați și accesibili.

Spre deosebire de software-ul desktop tradițional, Thin Clients bazați pe web nu au nevoie de procese importante de instalare și execuție pe dispozitivul utilizatorului. În schimb, prelucrarea și evaluarea datelor se efectuează în principal pe un server Web de la distanță și doar rezultatul procesării datelor este transmis la computerul client local al utilizatorului pentru afișare sau output, de obicei printr-un browser web care gestionează comunicarea cu serverul web (prin protocolul HTTP), precum și reprezentarea interfeței cu utilizatorul. În această secțiune, este prezentat un Thin Client web compatibil inter-pltformă, care utilizează noile specificații JavaScript și HTML5 cu interfața API pentru aplicații online "Service Worker" și Bootstrap Framework, prezentând arhitectura, mediul de dezvoltare și avantajele sale în gestionarea resurselor, administrarea informațiilor contextuale și securitate sporită.

6.5.1. Adaptarea conținutului la terminal

Mărimea și rezoluția afișată pe laptop, desktop, tabletă, smartphone și televizor poate varia considerabil. Din acest motiv, afișarea și funcționarea unui Thin Client bazat pe servicii web depinde foarte mult de cerințele/posibilitățile diferite ale dispozitivelor.

Scopul adaptării conținutului la terminale este acela de a-și modifica prezentarea astfel încât Thin Clients să se prezinte cât mai clar și în mod cât mai optimizat cu fiecare tip de vizualizare. Terminalul încarcă automat partea corectă a fișierului CSS și afișează conținutul în funcție de dimensiunea ecranului. Interfața cu utilizatorul utilizează servicii web personalizate pentru indicatoare grafice, contoare, casete de text, intrări și butoane, care sunt, de asemenea, adaptabile la diverse dispozitive, așa cum se arată în figura 39.

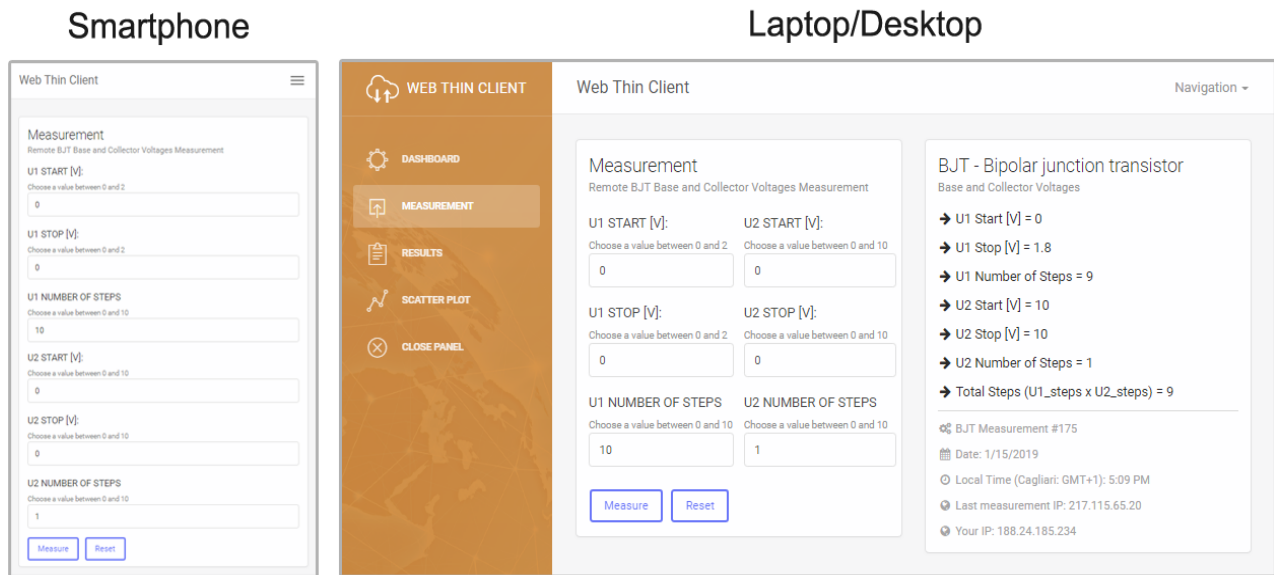


Figura 39: Afișarea de către Thin Client a tele-măsurări pentru o rezoluție de dispozitiv mobil (stânga) și afișaj laptop/desktop (dreapta).

Acesta este modul *sustenabil - durabil* în care a fost implementată adaptarea conținutului la terminal între diferite platforme, deoarece sunt utilizate numai tehnologii standardizate, iar software-ul va fi, de asemenea, compatibil cu dispozitivele ulterioare care vor fi construite folosind acest standard.

6.5.2. Management prin metodologia web progresivă

Folosind acest model de aplicație, au fost combinate posibilitățile oferite de cele mai moderne browsere cu beneficiile utilizării mobile. Termenul "progresiv" se referă la faptul că, din punctul de vedere al experienței utilizatorului, Thin Client-ul se adaptează în mod progresiv la dispozitiv și permite gestionarea și personalizarea mai multor caracteristici, inclusiv viteza de accesare și optimizarea dispozitivelor, care anterior erau rezervate numai pentru produsele software de tip nativ [191]. Au fost combinate, în acest fel, avantajele unui Thin Client clasic și al unei aplicații personalizate desktop/mobil. Pentru a implementa caracteristica descrisă anterior, s-a utilizat standardul HTML5, CSS3 și JavaScript.

În plus, "Service Workers" au fost folosiți pentru a gestiona optimizat memoria cache optim al funcționalităților offline. Un Service Worker este un proxy de rețea programabil, permițând controlul cererilor de rețea de la aplicație. Acesta își bazează execuția pe evenimente de acces personalizate. Protocolul HTTPS a fost utilizat pentru comunicarea securizată între clientul web și serverul web.

6.5.3. Capabilități offline

Cu ajutorul unui ServiceWorker [194], aplicația a fost configurată să utilizeze active prioritare din cache, permițând o experiență specifică a utilizatorilor online chiar înainte de încărcarea mai multor date din rețea.

Din punct de vedere tehnic, Service Worker furnizează un proxy de rețea implementat prin cod JavaScript în browserul web pentru a gestiona solicitările Web/HTTP dintr-un program, interpunându-se între conexiunea la rețea și terminalul care furnizează conținutul. În acest fel, mecanismele cache pot fi utilizate în mod eficace și permit comportamentul fără erori pe perioade lungi de utilizare offline. Arhitectura propusă este prezentată în figura 40.

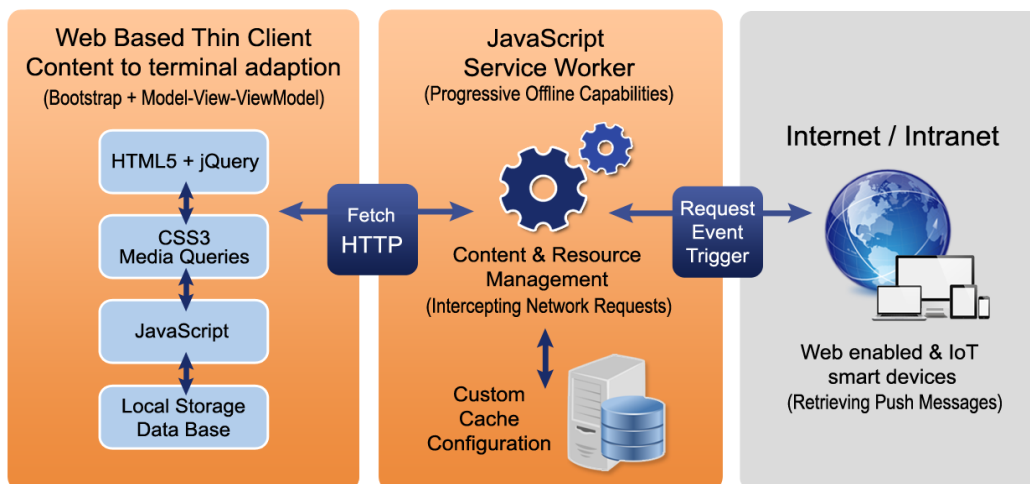


Figura 40: Arhitectura propusă pentru web Thin Client

Thin client-ul îndeplinește ulterior toate sarcinile care îi sunt atribuite în mod independent. Accesul offline este gestionat de Service Worker, care, odată instalat în browserul de navigație, interceptează cererile de rețea și efectuează acțiuni corespunzătoare în funcție de disponibilitatea rețelei.

Această caracteristică permite utilizatorului să acceseze resursele chiar și fără conexiune și, de asemenea, să îmbunătățească calitatea experienței (QoE) și calitatea serviciilor (QoS). Chiar dacă există o conexiune, unele fișiere nu vor trebui să fie încărcate de pe server deoarece au fost deja stocate local.

Pentru ca funcțiile JavaScript să poată fi executate, a fost întotdeauna o condiție prealabilă că resursa să fie deschisă într-un browser [195] - folosind această metodă, pe de altă parte, se permite executarea unui fișier JavaScript chiar dacă resursa asociată nu este deschisă deloc.

Script-ul are, de asemenea, opțiunea de a încărca conținut nou în fundal - de exemplu, un rezultat de la o măsurare efectuată anterior. Dacă datele sunt accesate ulterior, conținutul va fi deja disponibil, ca o copie de rezervă.

În plus, programul software funcționează numai pe conexiuni securizate HTTPS. Acest lucru are în primul rând motive de securitate, dar a fost construit și într-un mod de perspectivă evolutivă - *sustenabil* - deoarece este probabil ca noile standarde să impună pe viitor ca resursele "sensibile" să fie accesibile numai pe conexiuni HTTPS.

De asemenea, Service Worker rulează propriul *thread* și nu permite manipularea directă pentru DOM (Document Object Model) "părinte" - având interfața bazată pe mesaje web. Funcționează ca un controler, de tip proxy sau interceptor: are propriul cache și poate comuta între fiecare cerere de rețea de ieșire. Capacitatea *offline* este realizată printr-o decizie pre-configurată automat - dacă se poate solicita un răspuns din memoria cache sau este necesară transmiterea unei noi cereri către rețea.

6.6. Sumarul capitolului

Internetul obiectelor (IoT) oferă posibilitatea de a conecta la Internet miliarde de dispozitive, resurse și obiecte inteligente. Totuși, aceste dispozitive conectate sunt utilizate adesea pentru soluții cu aplicații limitate pentru o resursă determinată care prezintă o descriere semantică limitată la un domeniu specific. În acest capitol, am încercat să extind această abordare inițială a IoT, către o viziune mai deschisă, interoperabilă și mai colaborativă. Primul pas în această direcție a fost abordarea Web of Things (WoT). WoT face ca IoT să poată avea un nivel comun, de servicii web RESTful. Dar, chiar și atunci când se *omogenizează accesul* prin intermediul HTTP, nu se dobândește încă o *semnificație* unitară a datelor vehiculate în IoT - managementul interconectării este încă preponderent *orientat pe conexiune*.

Pentru un adevărat salt calitativ – un management IoT *orientat pe informație* – capitolul 6 prezintă conceptul de Web Semantic al Obiectelor (SWoT – Semantic Web of Things), care integrează semantica în IoT.

Web-ul Semantic al Obiectelor urmărește să ofere o mai mare interoperabilitate și un potențial pentru Big Data (datele "mari") pentru a reduce la minim incompatibilitățile din actualul "circuit al valorii" în IoT. Se deschid noi oportunități de piață, alimentate de capacitățile SWoT. Web of Things și, în consecință, SWoT ar putea fi principalele motoare ale *convergenței* IoT, deoarece aceleași protocoale existente (HTTP) și mijloacele de pe partea clientului ("client side"), de exemplu, browserele, pot fi *reutilizate* pentru a *furniza evenimente* (notificări, citire/scriere/actualizare de date) sau *invocări de servicii* (cerere/răspuns) la modul generalizat în Cloud.

Odată cu evoluția tehnologiei, sistemele de operare - SO (OS) au fost considerate "fără barieră" (de la micro-SO încorporate – "embedded tiny OS" la marile Centre de Date). Cu toate acestea, există mari diferențe de accesibilitate de la un sistem de operare la altul. În acest capitol am prezentat și am propus o soluție utilizând o tehnologie standardizată și compatibilă cu toate platformele moderne, care funcționează independent de sistemul de operare și de dispozitivele utilizate, extinzând deschiderea "fără barieră" pentru utilizatorii web și de telefonie mobilă. Se impune separarea strictă necesară a structurii unui document (Document Object Model) și a prezentării sale (layout) utilizând standardele HTML5, CSS3, JavaScript și WAI-ARIA, care produc o "interfață utilizator fără barieră", fără a compromite designul aplicațiilor și scalabilitatea acestora, făcându-le perceptibile și tangibile pentru toți utilizatorii.

Pentru o și mai mare *deschidere a metodelor de acces* (exemplificată în comunicațiile om-mașină dar direct extensibilă la comunicațiile mașină-mașină, în particular la IoT), am utilizat standarde web deschise (*open source*) în soluții asistive. Utilizatorii cu deficiențe de vedere nu se limitează la o formă tipică de tehnologie asistivă și pot alege ce tip de software (de exemplu cititor de ecran) sau hardware (de exemplu, cititor Braille) doresc să utilizeze. Aplicația dezvoltată cu marcaj semantic WAI-ARIA este compatibilă cu toate browserele moderne și cititoarele de ecran, fără a fi necesar să se instaleze plug-ins suplimentare sau module terțe, care pot provoca încălcări ale securității, pot genera consum de resurse inutile și pot afecta QoE (calitatea experienței) din cauza dependențelor suplimentare. Chiar dacă se utilizează un browser vechi sau o tehnologie de asistență care nu înțelege marcajul semantic utilizat, informațiile vor fi ignorate, deoarece WAI-ARIA a fost implementată ca o extensie pentru a îmbunătăți marcarea HTML spre a crește valoarea informațională intrinsecă fără a-l modifica perceptual (vizual-acustic-tactil).

Una dintre cele mai importante caracteristici ale soluției propuse este că toate componentele pot fi focalizate, actualizările continue ale arborelui de acces sunt transmise cititoarelor de ecran și afișajelor Braille, făcând informațiile accesibile utilizatorului fără a fi nevoie să recitească întreaga pagină după accesarea unui eveniment. Integrarea tehnologiei AMP (Accelerated Mobile Pages) în soluția prezentată, asigură compatibilitatea completă cu dispozitivele mobile, făcând aplicația accesibilă și optimizată pentru utilizatorii de smartphones și tablete.

Valoarea acestor contribuții rezidă, în special, la integrarea unor noi tehnologii ale web-ului semantic, accentul fiind pus pe conținut și rostul-semnificația lui. Cercetările menite a valoriza tehnologiile asistive (în contextul mai larg al clădirilor inteligente cu IoT integrat cognitiv) s-au înscris în Proiectul european de cercetare „NoAH – Not Alone At Home” – AAL-2015 „Active Assisted Living Programme” la care participă și universitatea noastră.

7. Concluzii generale și contribuții originale

Această secțiune prezintă principalele rezultate ale cercetării doctorale în *integrarea tehnologiilor web cu controlul IoT în timp real*.

Teza contribuie la *extinderea serviciilor web* de la nivelul OSI *prezentare* către nivelele inferioare (abordarea "de sus în jos" – "top-down").

În celălalt sens, către nivelele OSI superioare (abordarea "de jos în sus" – "bottom-up") sunt *expuse* (prin servicii web) *capabilități* ("controale") – "*controlul ca serviciu*". Prin intermediul *serviciilor REST*, se face o nouă abordare a *controlabilității* (*tranzițiile în automatul controlat prin stare al modelului comportamental IoT* sunt comandate prin cereri-metode HTTP).

Soluțiile propuse în lucrare se bazează pe noile tehnologii Web-RTC (comunicații web de timp real) – obiectivul a fost "*publicarea controlului*" (HTML etc.) pentru "*management IP*" prin *browser-e obișnuite*, pe *terminale accesibile* (chiar tablete sau telefoane inteligente – "smartphones"). Dat fiind că noile metode de *comunicații IP* – "over IP" au depășit deja viteza (respectiv constantele de timp de proces) și capacitatea (respectiv banda) impusă de cerințele industriale – ajungând să asigure și *comunicații critice* din punct de vedere al *securității cibernetice* ("cyber critical communications") – serviciile web pot prelua funcții de *control al comutației* în IoT, chiar până la "streaming" – comutația fluxurilor în IoT (în special în IoMT – Internet of Multimedia Things).

În lucrarea de față este extins conceptul *repartizării sarcinii computaționale* (inclusiv dublarea redundantă / "caching") Cloud computing / Edge computing de optimizare a localizării proceselor – centralizat / descentralizat. Acest demers a fost integrat în "agregarea obiectelor ca resurse" – alocarea "mobilizarea" acestora făcându-se în rețele "sociale". *Securitatea* reprezintă un important criteriu de alegere a strategiei de alocare a resurselor.

Strategia de alocare a resurselor urmărește ca prelucrarea și stocarea de volum mare să se facă localizat, la server și chiar la client obiectivul fiind un *minimum de transferuri la distanță* – unde se trimit mai puțin date și *mai mult comenzi* / semnalizări – sincronizări etc. O contribuție adusă în acest domeniu de către actuala lucrare constă în re-orientarea unor schimburi de date achiziționate, în sensul optimizării *produsului Procesare x Volum* la nivelul "Thin Client" (de exemplu SBC – Single Board Computer) prin trimiterea datelor achiziționate direct în Cloud unde se pot face calcule de mai mare anvergură (procesări digitale de semnal vectoriale, complexe, cu transformări integrale în domeniul timp și/sau în domeniul frecvență – precum cele implementate în proiectul "Con-Intel") urmând ca rezultatul (respectiv energia activă și reactivă contorizată) să fie returnat (de exemplu pentru simpla afișare – "citirea contorului") la nivelul client.

Chiar dacă metodele de acces la resurse care au fost consolidate în lucrare au pornit de la comunicarea om-mașină (H2M), *tehnicele de acces* dezvoltate sunt extensibile la comunicarea mașină-mașină (M2M), în particular la IoT. Așadar, abordarea *nivelelor de prezentare transcende* perspectiva utilizatorului uman, către o perspectivă unificată a *comunicațiilor IoT ca partajare a datelor și comenzilor*. Prin această extindere a limbajelor de *markup* de la mijloace de programare a legăturilor în IoT multi-modal la mijloace directe de "brokeraj", etichetele *markup* devin și adnotări semantice dar și attribute de control bazat pe evenimente sau informații utile pentru intermediere (inclusiv mediere semantică).

În contextul acestui demers integrativ - așa cum semnalizările/notificările "urcă" spre HTTP (ca protocol unificator) așa și expunerea de *capabilități* "urcă" spre nivelul *Prezentare* al stivei OSI astfel încât această lucrare contribuie la managementul diversității sistemelor ciber-fizice (CPS, "cyber-physical systems").

Prin extinderea limbajelor de *markup* – de la mijloace de programare a legăturilor în IoT multi-modal la mijloace de directe de intermediere semantică – “brokerajul” între dispozitive IoT atât de diverse ca protocoale și cozi de mesaje “urcă”, se contopește în nivelul Prezentare.

Viziunea mea și “deschiderea” acestei teze de doctorat este utilizarea viitoare a limbajelor de marcare pentru a programa direct gestiunea IoT, SIoT (Social IoT), WoS (Web of Sensors), WoT (Web of Things), SWoT (Semantic WoT) etc – într-o abordare cuprinzătoare (cum cuprinzătoare a fost și gestiunea conținutului, adaptarea la terminale, receptivitatea / reactivitatea serviciilor web).

În acest sens, soluțiile propuse și dezvoltate capătă *sustenabilitate și durabilitate* – accentul fiind pus pe o nouă viziune asupra *interfațării orientate pe informație, eliberată de constrângerile date de conexiuni / de rețea și, pe de-altă parte, eliberată și de constrângerile sintactice.*

7.1. Principalele contribuții originale

În contextul stratificării IoT, al controlului bazat pe evenimente și al orientării pe servicii pentru mediul distribuit și Cloud, dar și al integrării managementului semantic cu statisticile avansate de timp real:

- © Am dezvoltat un sistem de **analiză a datelor în timp real folosind servicii web, metadate semantice și procesare Cloud** (“real-time data analytics”), cu scopul de a obține performanțe îmbunătățite în vizibilitatea datelor și creșterea competitivității analitice, ceea ce ar putea facilita viitoare dezvoltări ale inteligenței artificiale în direcția raționamentului automat în timp real („real-time machine reasoning”). Deși tipurile de date accesate pe dispozitive variază foarte mult, tendința este către colectarea acestora în Centre de Date pentru analiza lor în timp real (sau “aproape în timp real”). Am arătat că productivitatea (în utilizare) și precizia conținutului expus în Internet pot fi sporite prin folosirea metadatelor web și a definirii semantice a *relației dintre părțile conținutului* – în contextul IoT se pot gestiona în timp real conceptele “sociale” de existență, prezență și disponibilitate. Soluția propusă are la bază împărțirea informațiilor în fragmente mici, slab conectate, folosind micro-formate, cu marcare dinamică bazată pe statistici de timp real „stream analytics”. [*descriere în capitolul 2, validare și diseminare în cadrul proiectului 7.3.5.4 și al articolului 7.3.3.3*].
- © Am contribuit la **integrarea limbajelor de marcare (“markup”) în managementul interfețelor semantice din IoT** cu scopul de a asigura un flux de lucru adecvat orientat nu atât pe conexiune cât pe specificul informației. Am utilizat tehnologii de ultimă generație, atât pe partea de “back-end” (“Server Side Scripting”, structuri de baze de date, biblioteci și module) cât și pe partea “front-end” (adnotare-etichetare semantică, tehnologii HTML, CSS, JavaScript și grafică web) pentru implementarea și mentenanța aplicațiilor, scalabile și compatibile cu platformele moderne, cât și pentru optimizarea timpului de programare, de administrare, și pentru utilizarea mai eficientă a resurselor [*descriere în capitolul 6, validare și diseminare în cadrul proiectelor 7.3.5.3 și 7.3.5.5 și al articolelor 7.3.2.1 și 7.3.3.2*].
- © Am integrat două soluții de aplicabilitate a **semanticii web pentru optimizarea metadatelor în comutația fluxurilor multimedia** – în cazul limitărilor de acces (la controale asociate, conținut ierarhizat, actualizare dinamică) respectiv în cazuri de conectivitate parțială.

Controlul pe bază de atribute fiind implementat în urmărirea comportamentală prin metadata și conținut semantic, provocarea privind managementul resurselor în IoT a fost preluată de „declanșatoare” (“triggers”) care gestionează consumul de resurse. În demonstratorul elaborat, optimizarea a fost orientată cu precădere către resursele multimedia, pentru managementul deciziilor de agregare a acestora și asociere în paradigma FaaS - *Function as a Service* [descriere în capitolul 6, validare și diseminare în cadrul proiectului 7.3.5.3 și al articolelor 7.3.3.2 și 7.3.2.1].

Următoarea contribuție are scopul de a îmbunătăți și a aduce plus-valoare metodologiei sus-menționate, venind în întâmpinarea tendinței evolutive a Internetului obiectelor și mediului distribuit, în care cantitatea de date transmise / publicate crește exponențial (spre “Big Data” – “masive de date”, “Data Mines”). Extinzând instrumentele care înlocuiesc utilizatorul uman – cu sarcina de a procesa și filtra informații pentru construirea, analiza și managementul de noi modele comportamentale – un pas către o lume în care “oamenii și mașinile au drepturi egale” în ceea ce privește utilizarea informațiilor și securizarea acestora l-a reprezentat:

- © Implementarea unui monitor de mesaje HTTP (Hypertext Transfer Protocol) și WebRTC (Web Real-Time Communication), analiza fiind bazată pe extragerea și filtrarea semantică a datelor multimedia de la senzori, înainte și după etichetarea corespunzătoare a acestora. Decizia de “instanțiere” pentru alocarea resurselor etichetate, inclusiv gestionarea resurselor ca serviciu s-a făcut prin cele mai noi metode “stateless” – “connectionless” – “serverless”, cu noi posibilități de scalare și optimizare, după criteriile de eficiență a costurilor și îmbunătățire a securității, al accesului și al autentificării, prin descentralizare, astfel încât *securizarea se transformă într-o soluție de control și management a obiectelor multimedia* [descriere în capitolul 4, validare și diseminare în cadrul proiectului 7.3.5.1 și al articolelor 7.3.1.6, 7.3.3.1 și 7.3.3.5].

Extinzând conceptul IoT la noua paradigmă SloT (“Social IoT”) – “obiecte sociale” aflate în “relații sociale” (D2D, peer-to-peer / multi-peer) – se urmărește ca oamenii să poată influența aceste reguli din SloT unde interacțiunile se desfășoară însă *autonom*. Pentru o separare și securizare a celor două “rețele sociale”, cea umană accesează doar rezultatele elaborate în cea instrumentală.

- © Am contribuit la **compoziția și interacțiunea dinamică a serviciilor IoT în mod social, bazată pe modelul “serviciu ca obiect”, și am proiectat și dezvoltat un demonstrator pe platforma IoT, pentru compoziția și abordarea serviciului ca obiect SWA – (“Socially Aware Object” - obiect conștient social).** În acest scop, am definit un model de analiză a serviciilor, am formulat un model de abstractizare a acestora, am definit modelele de interacțiune socială între serviciile înseși și între servicii și utilizatori, am integrat indicatorii de calitate tipici pentru domeniul de aplicare al SloT și optimizarea regulilor de căutare a serviciilor definite. Am integrat stratul de prezentare din stiva OSI în controlul și gestiunea obiectelor – astfel, dacă un obiect nu este găsit fizic se cooptează un “avatar” al acestuia (emulat, virtualizat) – într-o concepție unitară „soft-logic” bazată pe conceptele de interactivitate ca serviciu (*Interactivity as a Service*) și tranzacție ca serviciu (*Transaction as a Service*) [descriere în capitolul 3, validare și diseminare în cadrul proiectului 7.3.5.3 și al articolului 7.3.1.1].

Am sistematizat descrierea IoT pe straturi și pe declararea – “*expunerea*” de *capabilități*, de la nivelul inferior către cele superioare OSI. Dată fiind organizarea IoT pe nivele de interacțiune, am urmărit ca o parte din tranzacții să se poată face între entități virtuale - numai strictul necesar să fie făcut cu actualizare fizică, la nivelele inferioare OSI.

Astfel:

- © Am definit și propus o **metodologie de acces la resurse în Internetul obiectelor orientată pe management telematic și acces receptiv (“responsiv”)**. Am urmărit controlul unificat, compatibilitatea inter-dispozitive, reducerea puterii de calcul necesare și îmbunătățirea securității prin minimizarea (uneori chiar eliminarea) dependențelor de terți (furnizori de servicii) – dependențe care pot crește vulnerabilitatea și necesită măsuri suplimentare de protecție. În acest scop am luat în considerare *generalizarea conceptului de “apel”* (ca “apel la resurse” – umane sau instrumentale) incluzând *descoperirea acestora* (având la bază *expunerea capabilităților*) – printr-un ansamblu de servicii care se pot extinde / se pot scala pentru managementul personalizat al eliberării (de-alocării) resurselor angajate. Validarea metodologiei s-a făcut prin integrarea sistemului de servere web și servicii Cloud de la National Instruments, printr-un set de instrumente virtuale (VI) programate LabVIEW și accesate pe multiple terminale, atât în experimente telematice reale cât și simulate. S-a urmărit un nivel înalt de QoE (calitate a experienței) la nivelul utilizatorului – îmbunătățirea vizualizării, de exemplu prin adaptarea reprezentării grafice (a datelor obținute) la diferite rezoluții ale ecranului (“adaptarea conținutului la terminal”) [*descriere în capitolul 5, validare și diseminare în cadrul articolelor 7.3.1.5 și 7.3.1.6*].

Deoarece, în aceeași direcție de cercetare doctorală, am urmărit și evaluarea, integrarea și îmbunătățirea serviciilor web cu tehnologiile de virtualizare în controlul fluxurilor de date în timp real și informatica instrumentală, am elaborat o serie de soluții originale privind:

- © “**Tele-măsurarea cu instrumentație virtuală utilizând servicii web**”. Prin reprezentarea unificată a stărilor / evenimentelor (de tranziție) a fost posibilă o integrare mai bună a gestionării resurselor într-un mediu distribuit prin simplificarea mașinilor algoritmice de stare asociate atât cu serviciile, cât și cu modelele comportamentale ale instrumentelor. Variabilele clasice de tip *publish-subscribe* au fost gestionate acum prin servicii și variabile Web. Datele produse de senzori, *însoțite de metodele lor de prezentare*, au fost preluate dinamic și adaptate la “terminal”, acesta fiind programat pentru a gestiona și a conecta serviciile de tele-măsurare, utilizând o interfață grafică cu utilizatorul, prin acces simplu via browser (navigator web), fără a fi nevoie să se instaleze module și “plugins” suplimentare, de la terți. Așa cum se poate observa încă din figura 1, aceste soluții aparțin de cele mai avansate tendințe, la partea superioară a “Graficului evoluției IoT”. [*descriere în capitolul 5, validare și diseminare în cadrul articolului 7.3.1.6*].

- © Prezentarea și dezvoltarea unui demonstrator de **“tele-măsurare și agregare a datelor în Cloud”** cu scopul de a echilibra procesele Edge computing și Cloud computing, în rețelele inteligente de comunicații, prin capabilități de instrumentație la care accesul la resurse și controlul acestora a fost preluat de către servicii web în comunicații H2H (“Human-to-Human”), M2M (“Machine-to-Machine”), H2M, cu precădere în aspecte telematice, IoT și instrumentație virtuală [*descriere în capitolul 5, validare și diseminare în cadrul proiectului 7.3.5.2 și a articolului 7.3.1.5*].

Am extins abordarea clasică spre noile paradigme Internet of Multimedia Things (IoMT) și dispozitive IoT activate web (“web enabled IoT”) contribuind la integrarea lor cu conceptul “WoT” (Web of Things). Conceptul de „Thin Client” (client slab / subțire / simplificat – cu cerințe de procesare / stocare / comunicații reduse pe dispozitiv / la minimum cerut de aplicație) este extins în această lucrare, de la *prezentare* la *controlabilitate*, definind și prezentând astfel conceptul de “controlabilitate ca serviciu”. Astfel:

- © **Am expus capabilități instrumentale prin servicii web de mare accesibilitate (inclusiv de pe “Thin Clients”)**. Gradul înalt de *inter-operabilitate* (pe o mare diversitate de dispozitive, platforme și sisteme de operare) este dat de *soluția de acces și execuție* într-un navigator web (browser). Demonstratorul a fost testat pe servicii web de *tele-măsurări reale și emulate* [*descriere în capitolul 6, validare și diseminare în cadrul proiectului 7.3.5.2 și al articolelor 7.3.1.2 și 7.3.1.4*].

În acest sens am venit în întâmpinarea tendințelor moderne pe direcția *virtualizării* (prin servicii web) a *execuției și managementului* proceselor industriale, bazată pe *unificarea accesului* (“*ubicuu*” – independent de poziție și mobilitate, independent de distanță), a controlului *distribuirii resurselor* (concentrat sau descentralizat), a *reprezentării* (indiferent de *format* și cu facilități de *adaptare a conținutului la terminal* – “content to terminal adaption” – la dimensiunea *afișajului* dispozitivului, la metodele de *introducere / selectare* disponibile – ecran tactil “touch screen”, “mouse”, tastatură) și la *lățimea de bandă* disponibilă – având în vedere cerințele de calitate a serviciilor / a experienței (“Quality of Services / of Experience”) – QoS / QoE.

Am abordat una din direcțiile particulare la nivel Client care este – în domeniul IoT – conceptul de „*Tablou de bord*” (“*Dashboard*”). *Controlul pe bază de evenimente* (considerate ca “declanșatoare” ale tranzițiilor modelate REST) a fost implementat prin servicii web. În contextul masivelor de date (“Big Data”), în vederea analizei (“Data Analytics”), prelucrării și a folosirii acestora, nu se mai trimit datele la proces ci procesul la date. Așa cum am arătat mai sus, optimizarea și gestiunea resurselor se poate realiza prin ontologia web-ului semantic și generalizarea conceptului de *cache / buffer* în perspectiva utilizatorului (om/mașină) și a aplicabilității (tehnic/business) orientarea făcându-se către maximizarea raportului performanță/costuri:

- © Am implementat **managementul avansat al resurselor pentru “Thin Clients” folosind “capabilități offline progresive”**. Prin această contribuție am combinat avantajele metodelor de management tradiționale, *orientate pe conexiune*, cu cele *orientate pe informație* – pe specificul ei (utilizate la “Thin Clients” ce funcționează bazat pe servicii web) adăugându-le capabilitatea de a funcționa și offline – din punctul de vedere al utilizatorilor care să-și poată continua activitatea fără a fi afectați (în sensul evitării pierderilor de date / evitării pierderii sesiunii de lucru) atunci când *conexiunea este intermitentă sau lipsește pe durate mai mari*. Noutatea soluției propuse este operarea mai ușoară (uneori utilizatorii nici nu sunt notificați asupra unor probleme sporadice

de conectivitate), cu reducerea efectivă a costurilor – dată fiind gestionare eficientă a resurselor sistemelor de control centralizate sau descentralizate – într-un spectru extins de aplicații potențiale (de la telematică la centralizarea informațiilor de tip "rich-content" cu adaptare a conținutului la terminal). Managementul datelor de pe dispozitive devine mai ușor, iar analiza offline a datelor este beneficiul direct al acestei soluții. Demonstratorul de "Thin Client" realizat prin servicii web, are avantajul de a nu necesita instalarea pe dispozitiv – un avantaj foarte mare dată fiind tendința contrară, exagerată în ultimii ani, de înlocuire a serviciilor web cu "apps" (așa cum am arătat în Introducere) [descriere în capitolul 6, validare și diseminare în cadrul proiectului 7.3.5.3 și al articolului 7.3.2.1].

În comutația fluxurilor în timp real, datele se generează în mod incremental, multe dintre aceste date provenind de la diferite dispozitive IoT (senzori și traductoare inteligente etc.) astfel că majoritatea problemelor sunt legate de managementul conectării și autentificării securizate a acestor dispozitive la sistemele superioare ierarhic, pentru a fi mai ușor de gestionat (inclusiv în sensul scalabilității). În acest sens am realizat:

- © **Integrarea protocolul HTTP (Hypertext Transfer Protocol) și a metodelor REST, având ca punct de plecare izolarea pe nivele printr-o autentificare securizată; am stocat datele în serverul universității (fără nici o dependență de serviciile Cloud ale unor terți), adăugând stratul de management ca extensie, crescând valoarea adăugată prin opțiunile cheie Big Data: urmărire și analiză.** Toate aceste caracteristici, aduc Internetul obiectelor mai aproape de nevoile utilizatorilor, oferindu-i un control deplin asupra fluxului de date și accesibilității. Arhitectura sistemului dezvoltat are și avantajul a trei straturi complet independente: stratul de prezentare, stratul de management al datelor și serviciile Web NI (National Instruments) care includ un discriminator de adrese MAC. Fiecare strat "consumă" un API și "produce" un alt API [descriere în capitolul 5, validare și diseminare în cadrul proiectului 7.3.5.2 și al articolului 7.3.1.2].

În urma experienței acumulate în anii de cercetare doctorală am realizat două studii de anvergură - demersuri sumative care trasează și principalele direcții de viitor din domeniul tezei:

- © **Eliberarea și de-alocarea resurselor prin abordări sociale și tranzacționale, integrând concepte din rețele sociale („social networks”) capabil-prezent-disponibil.** "When Social Networks meet D2D Communications: A Survey", publicat într-un jurnal ISI din fluxul principal (Q1) cuprinde o trecere în revistă a convergenței dintre rețelele sociale și comunicarea "D2D" (Device to Device) așa cum este prezentată în literatura de specialitate. Conceptul a fost propus în urmă cu cinci-șase ani – numărul de lucrări, chiar dacă este limitat, este totuși relevant în acest interval de timp, arătând că acest subiect reprezintă în mod clar o tendință emergentă. Am analizat această convergență sub trei aspecte diferite: descoperirea și selecția "releelor", selecția modurilor de comunicare și alocarea și gestionarea resurselor. Obiectivele principale ale lucrării au fost de a investiga beneficiile integrării abordării sociale cu comunicarea D2D și de a oferi o imagine clară a eforturilor științifice actuale din domeniul comunicării D2D. Din documentele analizate, rezultă clar modul în care rețelele sociale pot îmbunătăți comunicările D2D sub diferite aspecte.

Cu toate acestea, eficiența soluțiilor propuse mai trebuie dovedită cu ajutorul unor noi cazuri reale de utilizare [*descriere în capitolul 3, validare și diseminare în cadrul proiectului 7.3.5.3 și al articolului 7.3.1.1*].

În perspectiva extinderii interfețelor funcționale care controlează dispozitivele, prin intermediul sistemelor de control al proceselor, "panourilor/tablorilor de bord" etc, am extins conceptul și optimizarea către oferirea și dezvoltarea de interfețe care să permită chiar aplicațiilor și obiectelor să îndeplinească aceleași funcții. Deoarece interfețele fizice ale dispozitivelor inteligente sunt adesea limitate atât din motive funcționale cât și estetice, pentru a evita componente excesive (din n interfețe, n-1 pot rămâne nefolosite la un moment dat) dar și pentru a economisi energie, orientarea propusă în Internetul Obiectelor este de a se folosi *interfațarea software* prin suport și compatibilitate Web. Această abordare deschide un mare potențial pentru designul industrial inovator.

- © "AMP versus Bootstrap - Accelerated Mobile Pages vs Bootstrap Responsive Web Design" – reprezintă o analiză cuprinzătoare a soluțiilor de dezvoltare optime pentru performanța web (inclusiv a contribuțiilor aduse de cercetarea doctorală) – mai ales când vine vorba despre dispozitivele mobile. Diferențierea soluțiilor pentru platforme fixe și mobile duce, de obicei, la complicații tehnice imense și costuri mari. Utilizarea tehnologiilor pentru adaptarea conținutului la terminal, reactivitate / "responsivitate" și "aplatizarea" interfețelor (reducerea diversificării ierarhice și a imbricării proceselor) au fost evidențiate ca măsuri de facilitare a gestiunii și de reducere a costurilor de întreținere în cazul mai multor platforme web diferite, optimizate pentru diverse categorii IoT. Aceste măsuri sunt foarte importante în "poziționarea online" (extensie a conceptului de vizibilitate în IoT) aflată în centrul strategiilor de management al resurselor și optimizare pentru motoarele de căutare [*descriere în capitolul 6, validare și diseminare în cadrul articolului 7.3.2.2*].

7.2. Direcții viitoare de cercetare

În urma cercetărilor aprofundate în această teză am identificat următoarele posibile direcții de cercetare care se înscriu în marile obiective comunitare (U.E.) și în strategia de cooperare academică cu industria – pe o serie de priorități cum ar fi "*transformarea digitală*" în domeniul economic:

- Extinderea conceptului client-server prin modelul de tratare a întreruperilor cu preluarea rutinei de servire, în timp real, direct din Cloud, amplificând optimizarea controlului pe bază de evenimente;
- Îmbunătățirea proceselor de analiză în timp real în IoT cu ajutorul învățării automate pentru o mai bună eficiență în înțelegerea tendințelor business, vizualizarea datelor și creșterea competitivității. Orientarea pe Web va continua să fie reprezentată de tendința schimbului de informații de a fi tot mai deschis și mai transparent;
- Integrarea tehnologiei *BlockChain* în ecosistemele IoT pentru a asigura securitatea și confidențialitatea datelor, printr-un sistem descentralizat și distribuit pentru a proteja aspectele private în cadrul ecosistemele IoT;
- Automatizarea integrării tehnologiilor semantice, bazate pe formalismul reprezentării datelor, în gestionarea eterogenității volumului, vitezei și volatilității datelor în Internetul obiectelor, pentru extinderea de soluții generice de inteligență artificială la scară globală.

7.3. Validarea și diseminarea rezultatelor științifice în publicații și proiecte de cercetare

7.3.1. Lucrări proprii indexate ISI

7.3.1. a) Articol într-un Jurnal din fluxul principal ISI (Q1)

- [1] Nitti, M.; **Stelea, G.A.**; Popescu, V.; Fadda, M. When Social Networks Meet D2D Communications: A Survey. *Sensors Journal* 2019, 19, 396; doi.org/10.3390/s19020396
1 citare în Jurnal ISI – IEEE ACCESS și 2 citări în reviste BDI

7.3.1. b) Articole în volumele indexate ISI ale unor conferințe internaționale

[2] **GA Stelea**, V Fernoaga, C Gavrilă, D Robu, "Web-Service Based Thin Client for Tele-Measurement", The International Scientific Conference eLearning and Software for Education, Volume 2, pp. 128-134, Carol I" National Defence University; DOI: 10.12753/2066-026X-18-088, 2018

[3] V Fernoagă, **GA Stelea**, C Gavrilă, F Sandu, "Intelligent Education Assistant Powered by Chatbots", The International Scientific Conference eLearning and Software for Education, Volume 2, pp. 376-383, Carol I" National Defence University, DOI: 10.12753/2066-026X-18-122, 2018. **1 citare în revistă BDI IEEE Explore.**

[4] V. P. Fernoaga, **G.A. Stelea**, A. Balan and F. Sandu, "OCR-based Solution for The Integration of Legacy And-Or Non-Electric Counters in Cloud Smart Grids," 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME), Iasi, 2018, pp. 398-403. doi: 10.1109/SIITME.2018.8599200.

[5] V. Fernoaga, **G.A. Stelea**, D. Robu, F. Sandu, "Communication Solutions for Power Measurement in the Cloud", 2018 International Conference on Communications (COMM), Bucharest, 2018, pp. 397-402. doi: 10.1109/ICComm.2018.8484758. **1 citare în revistă BDI - IEEE Explore.**

[6] D.N.Robu, P.V.Fernoagă, **G.A.Stelea**, F.Sandu, "Tele-measurement with virtual instrumentation using web-services", October 2017 DOI: 10.1109/SIITME.2017.8259932 Conference: 2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME) **2 citări în reviste BDI - IEEE Explore.**

[7] L. Sangeorzan, N.Enache-David, **G.A. Stelea**, D. Streitferdt, E. Rozova, "Some Considerations about Text Classification Systems using Weka", 30th International Business Information Management Association Conference (IBIMA), pp. 1129 - 1131, 8.-9. Nov. 2017, ISBN: 978-0-9860419-9-0, Madrid / Spain, 2017.

[8] S. Zamfir, B.A. Lupu, **G.A. Stelea**, and D.N. Robu. "Extending Linux Controllability of Bluetooth Low Energy Devices in the IoT". In: 2017 International Conference on Optimization of Electrical and Electronic Equipment & 2017 International Aegean Conference on Electrical Machines and Power Electronics. (Braşov, Romania, May 25–27, 2017). IEEE Xplore, 2017, pp. 875–880. isbn: 978-1-5090-4488-7. **1 citare în revistă BDI - IEEE Explore.**

7.3.2. Lucrări proprii în curs de indexare ISI

- [1] **G.A. Stelea**, V. Popescu, M. Murrioni, T. Balan and V. Fernoaga, „Enhanced Resource Management for Web Based Thin Clients Using Cross-Platform Progressive Offline Capabilities”, 14th EAI International Conference on Cognitive Radio Oriented Wireless Networks - CROWNCOM 2019 | Poznan, Poland
- [2] L. Sangeorzan, N. Enache-David, **G.A Stelea**, „AMP versus Bootstrap” 32nd IBIMA Conference:15-16 November 2018, Seville, Spain Vol. 32, ISBN: 978-0-9998551-1-9

7.3.3. Lucrări proprii indexate în alte BDI (EBSCO)

- [1] **GA Stelea**, C Gavrilă, S Zamfir, R Curpen, "Face Recognition for Education in the Cloud", - eLSE: International Scientific Conference - eLearning and Software for Education - 2017, Vol. 2, p181-188. 8p. - doi.org/10.12753/2066-026x-17-111 – Bucharest - April 2017;
- [2] **G.A. Stelea**, V. Fernoaga, C. Gavrilă, V. Popescu, M. Murrioni, "Mobile Accessible Rich Internet Web Application enhanced with AMP publishing technology", Review of the "Henri Coandă" Air Force Academy No.1 (39)/2019, DOI: 10.19062/1842-9238.2019.17.1.9;
- [3] **Stelea, G.A.**, Gavrilă, C., Fernoaga, V.P.: Real-Time Data Analytics with Semantic Web Metadata and Web Services – Bulletin of the Transilvania University of Braşov - VOL. 10 (59) No.2 - 2017 Series I – Engineering Sciences, ISSN 2065-2119 (Print), ISSN 2065-2127 (CD-ROM);
- [4] **GA Stelea**, R Curpen, S Surariu, A Papoi, "Peer-to-peer Real Time Cloud Communication Solutions", Review of the "Henri Coandă" Air Force Academy - No 2 (32) 2016 - DOI: 10.19062/1842-9238.2016.14.2.12;
- [5] N. Enache-David, L. Sangeorzan, **G.A. Stelea**, "Data Analysis – Between Theory and Practice", October 2017 DOI: 10.19062/1842-9238.2017.15.2.11, Review of the Air Force Academy No. 2. Pp. 85-92, Henri Coanda Air Force Academy;

7.3.4. Monografiile și capitole din cărți cu caracter științific

[1] **G.A.Stealea**, F.Sandu, L.Sangeorzan, "Programarea în Internet, mediu distribuit și Cloud" - Monografie, 437 Pagini - Octombrie 2017, Editura Universității Transilvania din Brașov, ISBN: 978-606-19-0949-0;

[2] **G.Stealea**, Ș.Șurariu, T.Bălan, "WebRTC - Comunicații prin web în timp real", Capitol din Monografia „Rețele Inteligente de Telecomunicații”, pp. 314-363, 2016, Editori: D.N.Robu, C.Costache. Editura Universității Transilvania din Brașov, ISBN: 978-606-19-0820-2

[3] C.Costache, **G.Stealea**, D.Robu, "Comunicații Cloud - Abordarea orientată pe servicii", Capitol din Monografia „Rețele Inteligente de Telecomunicații” pp. 266-313, 2016, Editori: D.N.Robu, C.Costache. Editura Universității Transilvania din Brașov, ISBN: 978-606-19-0820-2

[4] C.Costache, V.Cazacu, **G.Stealea**, "Software pentru Crearea de Servicii", Capitol din Monografia „Rețele Inteligente de Telecomunicații”, pp. 127-135, Editori: D.N.Robu, C.Costache. Editura Universității Transilvania din Brașov, ISBN: 978-606-19-0820-2

[5] L.Sangeorzan, **G.A.Stealea**, N.David-Enache, "Web Development Techniques for Applications and Websites" – Carte cu caracter didactic, Limba Engleză, 117 pagini, 2016 Editura Universității Transilvania din Brașov, ISBN: 978-606-19-0782-3

7.3.5. Proiecte de cercetare

[1] Proiectul european de cercetare „NoAH – Not Alone At Home” – AAL-2015 „Active Assisted Living” Programme - *articolul 7.3.1.8*

[2] Proiectul PN-III-P2-2.1-PTE-2016-0064 "CON-INTEL – Măsurarea consolidată și transmiterea parametrilor energetici spre punctele de colectare" - Finanțat de UEFISCDI - Program PNCDI III – Subprogramul " Creșterea competitivității economiei românești prin cercetare, dezvoltare și inovare – Transfer la operatorul economic" - *articolele 7.3.1.3, 7.3.1.5 și 7.3.1.6*

[3] Proiectul "RISCOPRIRSI - Design and development of a demonstrator for the dynamic composition of tourism services using an IoT platform " - program regional italian (INDUSTRIAL AND INFORMATION ENGINEERING - SSD ING - INF/03 - CALL SECT. 09/F2 - TELECOMMUNICATIONS) – coordonator: Departamentul de Inginerie Electrică și Electronică - Universitatea din Cagliari, Italia - *articolul 7.3.1.1*

[4] Proiectul UEFISCDI de Mobilitate pentru Cercetători - PN-III-P1-1.1-MC-2017-0768 – în cadrul Departamentului de Inginerie Electrică și Electronică (DIEE) al Universității din Cagliari, Italia - *articolul 7.3.1.4*

[5] Proiectul " MARINANOW" – program finanțat de Regiunea Autonomă Sardinia - Italia (POR FESR 2014-2020 - Asse 1, Azione 1.1.3) – coordonator: Departamentul de Inginerie Electrică și Electronică - Universitatea din Cagliari, Italia - *articolul 7.3.3.2*

Bibliografie

- [1] S. M. Babu, A. J. Lakshmi, and B. T. Rao. 2015. A study on cloud based Internet of Things: CloudIoT. In 2015 Global Conference on Communication Technologies (GCCT). 60–65. doi.org/10.1109/GCCT.2015.7342624
- [2] S. Ziegler, "Considerations on IPv6 scalability for the Internet of Things — Towards an intergalactic Internet," 2017 Global Internet of Things Summit (GloTS), Geneva, 2017, pp. 1-4. doi: 10.1109/GIOTS.2017.8016238
- [3] S. Ziegler, S. Nikoletsea, S. Krco, J. Rolim and J. Fernandes, "Internet of Things and crowd sourcing - a paradigm change for the research on the Internet of Things," 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, 2015, pp. 395-399. doi: 10.1109/WF-IoT.2015.7389087
- [4] S. Singh and N. Singh, "Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce," 2015 International Conference on Green Computing and Internet of Things (ICGCloT), Noida, 2015, pp. 1577-1581. doi: 10.1109/ICGCloT.2015.7380718
- [5] S. Narang, T. Nalwa, T. Choudhury and N. Kashyap, "An efficient method for security measurement in internet of things," 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 2018, pp. 319-323. doi: 10.1109/IC3IoT.2018.8668159
- [6] A. J. Jara, "Wearable Internet: Powering Personal Devices with the Internet of Things Capabilities," 2014 International Conference on Identification, Information and Knowledge in the Internet of Things, Beijing, 2014, pp. 7-7. doi: 10.1109/IIKI.2014.9
- [7] L. Celia and Y. Cungang, "(WIP) Authenticated Key Management Protocols for Internet of Things," 2018 IEEE International Congress on Internet of Things (ICIOT), San Francisco, CA, 2018, pp. 126-129. doi: 10.1109/ICIOT.2018.00024
- [8] M. A. Jarwar and Ilyoung Chong, "Exploiting IoT services by integrating emotion recognition in Web of Objects," 2017 International Conference on Information Networking (ICOIN), Da Nang, 2017, pp. 54-56. doi: 10.1109/ICOIN.2017.7899474
- [9] Y. Sun, Y. Xia, H. Song and R. Bie, "Internet of Things Services for Small Towns," 2014 International Conference on Identification, Information and Knowledge in the Internet of Things, Beijing, 2014, pp. 92-95. doi: 10.1109/IIKI.2014.27
- [10] P. Urien, "Blockchain IoT (BIoT): A New Direction for Solving Internet of Things Security and Trust Issues," 2018 3rd Cloudification of the Internet of Things (CloT), Paris, France, 2018, pp. 1-4. doi: 10.1109/CIOT.2018.8627112
- [11] A. Azzarà, S. Bocchino, P. Pagano, G. Pellerano, and M. Petracca. 2013. Middleware solutions in WSN: The IoT oriented approach in the ICSI project. In 2013 21st International Conference on Software, Telecommunications and Computer Networks - (SoftCOM 2013). 1–6. <https://doi.org/10.1109/SoftCOM.2013.6671886>
- [12] Z. B. Babovic, J. Protic, and V. Milutinovic. 2016. Web Performance Evaluation for Internet of Things Applications. IEEE Access 4 (2016), 6974–6992. <https://doi.org/10.1109/ACCESS.2016.2615181>
- [13] B. Dorsemayne, J. Gaulier, J. Wary, N. Kheir and P. Urien, "Internet of Things: A Definition & Taxonomy," 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies, Cambridge, 2015, pp. 72-77. doi: 10.1109/NGMAST.2015.71
- [14] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman. 1999. An efficient multicast protocol for content-based publish-subscribe systems. In Proceedings. 19th IEEE International Conference on Distributed Computing Systems (Cat. No.99CB37003). 262–272. <https://doi.org/10.1109/ICDCS.1999.776528>
- [15] S. Bandyopadhyay and A. Bhattacharyya. 2013. Lightweight Internet protocols for web enablement of sensors using constrained gateway devices. In 2013 International Conference on Computing, Networking and Communications (ICNC). 334–340. <https://doi.org/10.1109/ICNC.2013.6504105>
- [16] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog Computing and Its Role in the Internet of Things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing (MCC '12). ACM, New York, NY, USA, 13–16. <https://doi.org/10.1145/2342509.2342513>
- [17] C. Bormann, S. Lemay, H. Tschofenig, K. Hartke, B. Silverajan, and B. Raymor. 2018. CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets. RFC 8323. RFC Editor.
- [18] Alessio Botta, Walter de Donato, Valerio Persico, and Antonio Pescapé. 2014. On the Integration of Cloud Computing and Internet of Things. In Proceedings of the 2014 International Conference on Future Internet of Things and Cloud (FiCLOUD '14). IEEE Computer Society, Washington, DC, USA, 23–30. <https://doi.org/10.1109/FiCloud.2014.14>
- [19] N. De Caro, W. Colitti, K. Steenhaut, G. Mangino, and G. Reali. 2013. Comparison of two lightweight protocols for smartphone-based sensing. In 2013 IEEE 20th Symposium on Communications and Vehicular Technology in the Benelux (SCVT). 1–6. <https://doi.org/10.1109/SCVT.2013.6735994>
- [20] Francisco Carpio, Admela Jukan, Ana Isabel Martín Sanchez, Nina Amla, and Nicole Kemper. 2017. Beyond Production Indicators: A Novel Smart Farming Application and System for Animal Welfare. In Proceedings of the Fourth International Conference on Animal-Computer Interaction (ACI2017). ACM, New York, NY, USA, Article 7, 11 pages. <https://doi.org/10.1145/3152130.3152140>

- [21] A. P. Castellani, T. Fossati, and S. Loreto. 2012. HTTP-CoAP cross protocol proxy: an implementation viewpoint. In 2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012), Vol. Supplement. 1–6. <https://doi.org/10.1109/MASS.2012.6708523>
- [22] X. Che and S. Maag. 2013. A Passive Testing Approach for Protocols in Internet of Things. In 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing. 678–684. <https://doi.org/10.1109/GreenCom-iThings-CPSCOM.2013.124>
- [23] Samia Allaoua Chelloug and Mohamed A. El-Zawawy. 2017. Middleware for Internet of Things: Survey and Challenges. *Intelligent Automation & Soft Computing* 0, 0 (2017), 1–9. <https://doi.org/10.1080/10798587.2017.1290328>
- [24] Y. Chen and T. Kunz. 2016. Performance evaluation of IoT protocols under a constrained wireless access network. In 2016 International Conference on Selected Topics in Mobile Wireless Networking (MoWNeT). 1–7. <https://doi.org/10.1109/MoWNeT.2016.7496622>
- [25] W. Colitti, K. Steenhaut, N. De Caro, B. Buta, and V. Dobrota. 2011. Evaluation of constrained application protocol for wireless sensor networks. In 2011 18th IEEE Workshop on Local Metropolitan Area Networks (LANMAN). 1–6. <https://doi.org/10.1109/LANMAN.2011.6076934>
- [26] V. Fernoaga, G.A. Stelea, D. Robu, F. Sandu, "Communication Solutions for Power Measurement in the Cloud", 2018 International Conference on Communications (COMM), Bucharest, 2018, pp. 397-402. doi: 10.1109/ICComm.2018.8484758.
- [27] N. Correia, D. Sacramento, and G. Schajitz. 2016. Dynamic Aggregation and Scheduling in CoAP/Observe-Based Wireless Sensor Networks. *IEEE Internet of Things Journal* 3, 6 (Dec 2016), 923–936. <https://doi.org/10.1109/JIOT.2016.2517120>
- [28] D. Dragomir, L. Gheorghe, S. Costea, and A. Radovici. 2016. A Survey on Secure Communication Protocols for IoT Systems. In 2016 International Workshop on Secure Internet of Things (SIoT). 47–62. <https://doi.org/10.1109/SIoT.2016.012>
- [29] J. L. Fernandes, I. C. Lopes, J. J. P. C. Rodrigues, and S. Ullah. 2013. Performance evaluation of RESTful web services and AMQP protocol. In 2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN). 810–815. <https://doi.org/10.1109/ICUFN.2013.6614932>
- [30] M. FRUSTACI, P. PACE, G. ALOI, and G. FORTINO. 2017. Evaluating critical security issues of the IoT world: Present and Future challenges. *IEEE Internet of Things Journal* PP, 99 (2017), 1–1. <https://doi.org/10.1109/JIOT.2017.2767291>
- [31] G. Farabaugh G. Pardo-Castellote and R. Warren. 2005. An Introduction to DDS and Data-Centric Communications. *Real-Time Innovations* August (2005).
- [32] J. Y. Huang, P. H. Tsai, and I. E. Liao. 2017. Implementing publish/subscribe pattern for CoAP in fog computing environment. In 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). 175–180. <https://doi.org/10.1109/IEMCON.2017.8117183>
- [33] C. Huo, T. C. Chien, and P. H. Chou. 2014. Middleware for IoT-Cloud Integration Across Application Domains. *IEEE Design Test* 31, 3 (June 2014), 21–31. <https://doi.org/10.1109/MDAT.2014.2314602>
- [34] M. Iglesias-Urkiá, A. Orive, M. Barcelo, A. Moran, J. Bilbao, and A. Urbieta. 2017. Towards a lightweight protocol for Industry 4.0: An implementation based benchmark. In 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM). 1–6. <https://doi.org/10.1109/ECMSM.2017.7945894>
- [35] Y. Jia, E. Bodanese, C. Phillips, J. Bigham, and R. Tao. 2014. Improved reliability of large scale publish/subscribe based MOMs using model checking. In 2014 IEEE Network Operations and Management Symposium (NOMS). 1–8. <https://doi.org/10.1109/NOMS.2014.6838311>
- [36] Jing, Athanasios Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. 2014. Security of the Internet of Things: Perspectives and challenges. *20 (11 2014)*, 2481–2501.
- [37] J. Joshi, V. Rajapriya, S. R. Rahul, P. Kumar, S. Polepally, R. Samineni, and D. G. K. Tej. 2017. Performance enhancement and IoT based monitoring for smart home. In 2017 International Conference on Information Networking (ICOIN). 468–473. <https://doi.org/10.1109/ICOIN.2017.7899537>
- [38] C. Lerche, N. Laum, F. Gólatowski, D. Timmermann, and C. Niedermeier. 2012. Connecting the web with the web of things: lessons learned from implementing a CoAP-HTTP proxy. In 2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012), Vol. Supplement. 1–8. <https://doi.org/10.1109/MASS.2012.6708525>
- [39] C. Lesjak, D. Hein, M. Hofmann, M. Maritsch, A. Aldrian, P. Priller, T. Ebner, T. Ruprecht, and G. Pregartner. 2015. Securing smart maintenance services: Hardware-security and TLS for MQTT. In 2015 IEEE 13th International Conference on Industrial Informatics (INDIN). 1243–1250. <https://doi.org/10.1109/INDIN.2015.7281913>
- [40] S. Zamfir, B.A. Lupu, G.A. Stelea, and D.N. Robu. "Extending Linux Controllability of Bluetooth Low Energy Devices in the IoT". In: 2017 International Conference on Optimization of Electrical and Electronic Equipment & 2017 International Aegean Conference on Electrical Machines and Power Electronics. (Brasov, Romania, May 25–27, 2017). IEEE Xplore, 2017, pp. 875–880. isbn: 978-1-5090-4488-7.
- [41] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiq, and I. Yaqoob. 2017. Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges. *IEEE Access* 5 (2017), 5247–5261. <https://doi.org/10.1109/ACCESS.2017.2689040>

- [42] X. Masip-Bruin, E. Marañón-Tordera, G. Tashakor, A. Jukan, and G. J. Ren. 2016. Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems. *IEEE Wireless Communications* 23, 5 (October 2016), 120–128. <https://doi.org/10.1109/MWC.2016.7721750>
- [43] H. V. Nguyen and L. L. Iacono. 2015. REST-ful CoAP Message Authentication. In 2015 International Workshop on Secure Internet of Things (SIoT). 35–43. <https://doi.org/10.1109/SIoT.2015.8>
- [44] G. Peralta, M. Iglesias-Urkia, M. Barcelo, R. Gomez, A. Moran, and J. Bilbao. 2017. Fog computing based efficient IoT scheme for the Industry 4.0. In 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM). 1–6. <https://doi.org/10.1109/ECMSM.2017.7945879>
- [45] T. Pflanzner and A. Kertesz. 2016. A survey of IoT cloud providers. In 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). 730–735. <https://doi.org/10.1109/MIPRO.2016.7522237>
- [46] J. Ramirez and C. Pedraza. 2017. Performance analysis of communication protocols for Internet of things platforms. In 2017 IEEE Colombian Conference on Communications and Computing (COLCOM). 1–7. <https://doi.org/10.1109/ColComCon.2017.8088198>
- [47] D.N.Robu, P.V.Fernoagă, G.A.Stealea, F.Sandru, “Tele-measurement with virtual instrumentation using web-services”, October 2017 DOI: 10.1109/SIITME.2017.8259932 Conference: 2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME)
- [48] S. Raza, D. Trabalza, and T. Voigt. 2012. 6LoWPAN Compressed DTLS for CoAP. In 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems. 287–289. <https://doi.org/10.1109/DCOSS.2012.55>
- [49] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke. 2016. Middleware for Internet of Things: A Survey. *IEEE Internet of Things Journal* 3, 1 (Feb 2016), 70–95. <https://doi.org/10.1109/JIOT.2015.2498900>
- [50] E. Rescorla and N. Modadugu. 2012. Datagram Transport Layer Security Version 1.2. RFC 6347. RFC Editor. <http://www.rfc-editor.org/rfc/rfc6347.txt>
- [51] M. Saleh, M. A. Abdou, and M. Aboulhassan. 2016. Assessing the use of IP network management protocols in smart grids. In 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA). 1–6. <https://doi.org/10.1109/AICCSA.2016.7945723>
- [52] T. Savolainen, N. Javed, and B. Silverajan. 2014. Measuring energy consumption for RESTful interactions in 3GPP IoT nodes. In 2014 7th IFIP Wireless and Mobile Networking Conference (WMNC). 1–8. <https://doi.org/10.1109/WMNC.2014.6878863>
- [53] C. Severance. 2015. Roy T. Fielding: Understanding the REST Style. *Computer* 48, 6 (June 2015), 7–9. <https://doi.org/10.1109/MC.2015.170>
- [54] Z. Shelby, K. Hartke, and C. Bormann. 2014. The Constrained Application Protocol (CoAP). RFC 7252. RFC Editor. <http://www.rfc-editor.org/rfc/rfc7252.txt>
- [55] A. B. Sulaeman, F. A. Ekadiyanto, and R. F. Sari. 2016. Performance evaluation of HTTP-CoAP proxy for wireless sensor and actuator networks. In 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob). 68–73. <https://doi.org/10.1109/APWiMob.2016.7811451>
- [56] S. N. Swamy, D. Jadhav, and N. Kulkarni. 2017. Security threats in the application layer in IOT applications. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). 477–480. <https://doi.org/10.1109/I-SMAC.2017.8058395>
- [57] N. Tantitharanukul, K. Osathanukul, K. Hantrakul, P. Pramokchon, and P. Khoenkaw. 2017. MQTT-Topics Management System for sharing of Open Data. In 2017 International Conference on Digital Arts, Media and Technology (ICDAMT). 62–65. <https://doi.org/10.1109/ICDAMT.2017.7904935>
- [58] D. Thangavel, X. Ma, A. Valera, H. X. Tan, and C. K. Y. Tan. 2014. Performance evaluation of MQTT and CoAP via a common middleware. In 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP). 1–6. <https://doi.org/10.1109/ISSNIP.2014.6827678>
- [59] P. Thota and Y. Kim. 2016. Implementation and Comparison of M2M Protocols for Internet of Things. In 2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science Engineering (ACIT-CSII-BCD). 43–48. <https://doi.org/10.1109/ACIT-CSII-BCD.2016.021>
- [60] H. Nishida, T. Nguyen, Optimal Client-Server Assignment for Internet Distributed Systems, *IEEE Transactions on Parallel and Distributed Systems*, Issue No. 03 - March (2013 vol. 24), pp 565-575.
- [61] N. Georgantas, G. Bouloukakis, S. Beauche, V. Issarny, Service-Oriented Distributed Applications in the Future Internet: The Case for Interaction Paradigm Interoperability, *Service-Oriented and Cloud Computing, ESOC 2013*, pp 134-148.
- [62] G. Pallis, T. Nguyen, *Cloud Computing: The New Frontier of Internet Computing*, *IEEE Internet Computing* (Volume: 14, Issue: 5, Sept.-Oct. 2010), pp 70 - 73.
- [63] R. Moreno-Vozmediano, R.S. Montero, I.M. Llorente, Key Challenges in Cloud Computing: Enabling the Future Internet of Services, *IEEE Internet Computing* (Volume: 17, Issue: 4, July-Aug. 2013), pp 18-25.
- [64] K. Gurparkash, S. Jaiteg, S.K. Maninderjit; I.M. Llorente, Semantic web services in clouds for semantic computing, *ACAI '11 Proceedings of the International Conference on Advances in Computing and Artificial Intelligence*, pp 229-232.
- [65] Stelea, G.A., Gavrilă, C., Fernoaga, V.P.: Real-Time Data Analytics with Semantic Web Metadata and Web Services - BULLETIN OF THE TRANSILVANIA UNIVERSITY OF BRASOV - VOL. 10 (59) No.2 - 2017 SERIES I

- ENGINEERING SCIENCES, ISSN 2065-2119 (Print), ISSN 2065-2127 (CD-ROM); P. Hitzler, F. van Harmelen, A reasonable Semantic Web, *Journal: Semantic Web*, vol. 1, no. 1,2 - 2010, pp 39-44.
- [66] J. Hendler, T. Berners-Lee, From the Semantic Web to social machines: A research challenge for AI on the World Wide Web, *Artificial Intelligence Volume 174*, Issue 2, February 2010, pp 156-161.
- [67] A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, S. Decker, Searching and browsing Linked Data with SWSE: The Semantic Web Search Engine, *Science, Services and Agents on the World Wide Web*, Vol. 9, 2011, pp 365-401.
- [68] W. Loibl, D. Tüz, A Methodology for Building Microformats, *Information and Communication Technologies in Tourism 2012*. Springer, Vienna, pp 108-119.
- [69] R. Meusel, P. Petrovski, C. Bizer, The WebDataCommons Microdata, RDFa and Microformat Dataset Series, In: Mika P. et al. (eds) *The Semantic Web – ISWC 2014*. Lecture Notes in Computer Science, vol 8796. Springer, Cham, pp 277-292.
- [70] T. Daisy Premila Bai, S. Albert Rabara, "Design and Development of Integrated Secured and Intelligent Architecture for Internet of Things and Cloud Computing", *Future Internet of Things and Cloud (FiCloud) 2015 3rd International Conference on*, pp. 817-822, 2015.
- [71] Atzori, L.; Iera, A.; Morabito, G.; Nitti, M. The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Comput. Netw.* 2012, 56, 3594–3608.
- [72] Pilloni, V.; Navaratnam, P.; Vural, S.; Atzori, L.; Tafazolli, R. Cooperative task assignment for distributed deployment of applications in wsns. In *Proceedings of the 2013 IEEE International Conference on Communications (ICC)*, Budapest, Hungary, 9–13 June 2013; pp. 2229–2234.
- [73] Colistra, G.; Pilloni, V.; Atzori, L. Task allocation in group of nodes in the IoT: A consensus approach. In *Proceedings of the 2014 IEEE International Conference on Communications (ICC)*, Sydney, NSW, Australia, 10–14 June 2014; pp. 23848–23853.
- [74] Chen, X.; Proulx, B.; Gong, X.; Zhang, J. Social trust and social reciprocity based cooperative D2D communications. In *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. Bangalore, India, 29 July–1 August 2013; pp. 2187–2196.
- [75] Du, Q.; Song, H.; Zhu, X. Social-feature enabled communications among devices towards smart IoT community. *arXiv 2018*, arXiv:1807.03590.
- [76] Zhang, Y.; Song, L.; Saad, W.; Dawy, Z.; Han, Z. Exploring social ties for enhanced device-to-device communications in wireless networks. In *Proceedings of the 2013 IEEE Global Communications Conference (GLOBECOM)*, Atlanta, GA, USA, 9–13 December 2013; pp. 4597–4602.
- [77] Wang, L.; Tang, H.; Cierny, M. Device-to-device link admission policy based on social interaction information. *IEEE Trans. Veh. Technol.* 2015, 64, 4180–4186.
- [78] Zhang, Y.; Pan, E.; Song, L.; Saad, W.; Dawy, Z.; Han, Z. Social network aware device-to-device communication in wireless networks. *IEEE Trans. Wirel. Commun.* 2015, 14, 177–190.
- [79] Zhang, B.; Li, Y.; Jin, D.; Hui, P.; Han, Z. Social-aware peer discovery for D2D communications underlying cellular networks. *IEEE Trans. Wirel. Commun.* 2015, 14, 1528–1539.
- [80] Ma, T.; Zhou, J.; Tang, M.; Tian, Y.; Al-Dhelaan, A.; Al-Rodhaan, M.; Lee, S. Social network and tag sources based augmenting collaborative recommender system. *IEICE transactions on Information and Systems* 2015, 98, 902–910.
- [81] Wang, R.; Yang, H.; Wang, H.; Wu, D. Social overlapping community-aware neighbor discovery for D2D communications. *IEEE Wirel. Commun.* 2016, 23, 28–34. [CrossRef]
- [82] L. Sangeorzan, N. Enache-David, G.A. Stelea, D. Streiferdt, E. Rozova, "Some Considerations about Text Classification Systems using Weka", 30th International Business Information Management Association Conference (IBIMA), pp. 1129 - 1131, 8.-9. Nov. 2017, ISBN: 978-0-9860419-9-0, Madrid / Spain, 2017.
- [83] R. T. Tiburski, L. A. Amaral, E. de Matos, D. F. G. de Azevedo, and F. Hessel. 2017. Evaluating the use of TLS and DTLS protocols in IoT middleware systems applied to E-health. In 2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC). 480–485. <https://doi.org/10.1109/CCNC.2017.7983155>
- [84] R. T. Tiburski, L. A. Amaral, E. D. Matos, and F. Hessel. 2015. The importance of a standard security architecture for SOA-based IoT middleware. *IEEE Communications Magazine* 53, 12 (Dec 2015), 20–26. <https://doi.org/10.1109/MCOM.2015.7355580>
- [85] Zhang, H.; Wang, Z.; Du, Q. Social-Aware D2D Relay Networks for Stability Enhancement: An Optimal Stopping Approach. *IEEE Trans. Veh. Technol.* 2018, 67, 8860–8874.
- [86] Zhang, M.; Chen, X.; Zhang, J. Social-aware relay selection for cooperative networking: An optimal stopping approach. In *Proceedings of the 2014 IEEE International Conference on Communications (ICC)*, Sydney, NSW, Australia, 10–14 June 2014; pp. 2257–2262.
- [87] Chiti, F.; Fantacci, R.; Pierucci, L. Social-Aware Relay Selection for Cooperative Multicast Device-to-Device Communications. *Future Internet* 2017, 9, 92.
- [88] VerneMQ. 2017. VerneMQ Broker. (2017). Retrieved March 1, 2018 from <https://vernemq.com/>
- [89] H. Wang, D. Xiong, P. Wang, and Y. Liu. 2017. A Lightweight XMPP Publish/Subscribe Scheme for Resource-Constrained IoT Devices. *IEEE Access* 5 (2017), 16393–16405. <https://doi.org/10.1109/ACCESS.2017.2742020>
- [90] Y. Xu, V. Mahendran, W. Guo, and S. Radhakrishnan. 2017. Fairness in fog networks: Achieving fair throughput performance in MQTT-based IoTs. In 2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC). 191–196. <https://doi.org/10.1109/CCNC.2017.7983104>

- [91] Militano, L.; Orsino, A.; Araniti, G.; Nitti, M.; Atzori, L.; Iera, A. Trusted D2D-based data uploading in in-band narrowband-IoT with social awareness. In Proceedings of the 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Valencia, Spain, 4–8 September 2016; pp. 21–26.
- [92] Militano, L.; Orsino, A.; Araniti, G.; Nitti, M.; Atzori, L.; Iera, A. Trust-based and social-aware coalition formation game for multihop data uploading in 5G systems. *Comput. Netw.* 2016, 111, 141–151.
- [93] Nitti, M.; Stelea, G.A.; Popescu, V.; Fadda, M. When Social Networks Meet D2D Communications: A Survey. *Sensors Journal* 2019, 19, 396; doi.org/10.3390/s19020396
- [94] J. Yang, K. Sandstrom, T. Nolte, and M. Behnam. 2012. Data Distribution Service for industrial automation. In Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies Factory Automation (ETFA 2012). 1–8. <https://doi.org/10.1109/ETFA.2012.6489544>
- [95] Jiang, J.; Zhang, S.; Li, B. Maximized Cellular Traffic Offloading via Device-to-Device Content Sharing. *IEEE J. Sel. Areas Commun.* 2016, 34, 82–91.
- [96] Bai, B.; Wang, L.; Wang, Z.; Chen, W.; Svensson, T. Caching based socially-aware D2D communications in wireless content delivery networks: A hypergraph framework. *IEEE Wirel. Commun.* 2016, 23, 78–81.
- [97] Duc-Hung Le, Nanjangud Narendra, Hong-Linh Truong, "HINC - Harmonizing Diverse Resource Information across IoT Network Functions and Clouds", Future Internet of Things and Cloud (FiCloud) 2016 IEEE 4th International Conference on, pp. 317-324, 2016.
- [98] Mohamed Abderrahim, Meryem Ouzzif, Karine Guilloard, Jérôme François, Adrien Lebre, Charles Prud'homme, Xavier Lorca, "Efficient Resource Allocation for Multi-Tenant Monitoring of Edge Infrastructures", Parallel Distributed and Network-Based Processing (PDP) 2019 27th Euromicro International Conference on, pp. 158-165, 2019.
- [99] Haodong Wang, Chiu C. Tan, Qun Li, "Snoogle: A search engine for pervasive environments", *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 8, pp. 1188-1202, 2010.
- [100] Shadi Aljawarneh, Muneer Bani Yassein, "A resource-efficient encryption algorithm for multimedia big data", *Multimedia Tools and Applications*, pp. 1-22, 2017.
- [101] Salvatore Loreto, Simon Pietro Romano, Real-Time Communication with WebRTC, O'Reilly Media, Inc., 2014, ISBN: 978-1-449-37187-6;
- [102] Alan B. Johnston, Daniel C. Burnett, WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web, Digital Codex LLC, 2014, ISBN: 978-0985978860;
- [103] Ruben Picek, Samuel Picek, WebRTC Multipoint Web Real-Time Communication, Springer International Publishing, 2014, ISBN: 978-3-319-07214-2;
- [104] GA Stelea, C Gavrilă, S Zamfir, R Curpen, "Face Recognition for Education in the Cloud ", - eLSE: International Scientific Conference - eLearning and Software for Education - 2017, Vol. 2, p181-188. 8p. - doi.org/10.12753/2066-026x-17-111 - Bucharest - April 2017 Siani Pearson, George Yee, Privacy and Security for Cloud Computing, Springer-Verlag London, 2013, ISBN: 978-1-4471-5793-9;
- [105] Sorin Zamfir, Titus Bălan, Florin Sandu, Cosmin Costache - Mobile communication solutions for the services in the Internet of Things - Proceedings of the 7th International Conference on Exploring Service Science, IESS 1.6 - Springer International, 2016 - pp 619-634, ISBN: 978-3-319-32688-7
- [106] Mike Cantelon, Marc Harter, T.J. Holowaychuk, Nathan Rajlich, Node.js in Action, Manning Publications Co., 2013, ISBN: 9781617290572;
- [107] Azat Mardan, Practical Node.js: Building Real-World Scalable Web Apps, Apress, 2014, ISBN: 978-1430265955;
- [108] Wu, D.; Zhou, L.; Cai, Y. Social-aware rate based content sharing mode selection for D2D content sharing scenarios. *IEEE Trans. Multimed.* 2017, 19, 2571–2582.
- [109] John D. Kelleher, Brian Mac Namee, Fundamentals of Machine Learning for Predictive Data Analytics, MIT Press. 2016 - ISBN: 978-0262029445
- [110] Stan Z. Li, Anil Jain, Handbook of Face Recognition, Springer, 2nd ed. 2011 - ISBN: 978-0857299314
- [111] Shai Shalev-Shwartz, Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014 - ISBN: 9781107057135
- [112] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016 - ISBN: 978-0262035613
- [113] Nishant Shukla, Machine Learning with TensorFlow, MEAP - Manning Publications, 2016 - ISBN: 9781617293870
- [114] GA Stelea, R Curpen, S Surariu, A Papoi, "Peer-to-peer Real Time Cloud Communication Solutions", Review of the "Henri Coanda" Air Force Academy - No 2 (32) 2016 - DOI: 10.19062/1842-9238.2016.14.2.12
- [115] D.N.Robu, C.Costache, "WebRTC - Comunicatii prin web în timp real", Capitol din Monografia „Rețele Inteligente de Telecomunicații”, pp. 314-363, 2016,. Editura Universității Transilvania din Brașov, ISBN: 978-606-19-0820-2
- [116] "WebRTC 1.0: Real-Time Communication between Browsers", W3C Working Draft, Aug. 2012, [online] Available: <http://www.w3.org/TR/webrtc/>.
- [117] Faliu Yi, Youhyun Kim, Inkyu Moon, "Secure Image-Authentication Schemes With Hidden Double Random-Phase Encoding", *Access IEEE*, vol. 6, pp. 70113-70121, 2018.
- [118] H. Schulzrinne, "RTP: A Transport Protocol for Real-Time Applications", IETF RFC 3550, 2003.
- [119] Titus C.Balan, Network policy function virtualization via SDN and packet processing - Review of the Air Force Academy, Brasov – RO, 2014

- [120] D. Wing, "Symmetric RTP / RTP Control Protocol (RTCP)", IETF RFC 4961, 2007.
- [121] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator Traversal for Offer/Answer Protocols", IETF RFC 5245, 2010.
- [122] J. Hautakorpi, "Requirements from Session Initiation Protocol (SIP) Session Border Control (SBC) Deployments", IETF RFC 5853, 2010.
- [123] T. Reddy, "STUN Extensions for Authenticated Firewall Traversal", IETF-Draft draft-reddy-rtcweb-stun-auth-fw-traversal.
- [124] R. Mahy, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", IETF RFC 5766, 2010.
- [125] Yi-Wen Luo, "The effect of using JavaScript and HTML to implement an E-learning web sites as a remedial teaching aid on English underachievers at universities of science and technology", Computers and Industrial Engineering (CIE) 2010 40th International Conference on, pp. 1-7, 25-28 July 2010.
- [126] W. Elleuch, "Models for multimedia conference between browsers based on WebRTC", Wireless and Mobile Computing Networking and Communications (WiMob) 2013 IEEE 9th International Conference on, pp. 279-284, 7-9 Oct. 2013.
- [127] Adham Zeidan, Armin Lehmann, Ulrich Trick, "WebRTC enabled multimedia conferencing and collaboration solution", WTC 2014; World Telecommunications Congress 2014; Proceedings of, pp. 1-6, 1-3 June 2014
- [128] Sredojev Branislav, Samardzija Dragan, Posarac Dragan, "WebRTC technology overview and signaling solution design and implementation", Information and Communication Technology electronics and Microelectronics (MIPRO) 2015 38th International Convention on, vol. 1006, no. 1009, 25-29 May 2015.
- [129] M. Phankokkrud, P. Jaturawat, "An evaluation of technical study and performance for real-time face detection using Web Real-Time Communication", Computer Communications and Control Technology (I4CT) 2015 International Conference on, pp. 162-166, 21-23 April 2015.
- [130] Jian Shen, Anxi Wang, Chen Wang, Patrick C. K. Hung, Chin-Feng Lai, "An Efficient Centroid-Based Routing Protocol for Energy Management in WSN-Assisted IoT", Access IEEE, vol. 5, pp. 18469-18479, 2017.
- [131] R. Fielding - REST: Architectural Styles and the Design of Network-based Software Architectures - University of California, Irvine, Doctoral dissertation, 2000
- [132] P. Cotfas; D. Cotfas; C. Gerigan - Simulated, hands-on & remote laboratories for studying the solar cells - Intl Conf on Optimization of Electrical & Electronic Equipment (OPTIM 2015), Brasov, Romania
- [133] I.Iliescu, T.Balan, O.Garoiu, S.Zamfir - Thin Client for Real -Time Monitoring of Communication Infrastructure - Intl Conf on Scientific Research and Education in the Air Force-AFASES 2016, Brasov, Romania
- [134] D. Guinard, I. Ion, and S. Mayer "In Search of an Internet of Things Service Architecture: REST or WS-*? A Developers' Perspective" - <https://www.vs.inf.ethz.ch/publ/papers/dguinard-rest-vs-ws.pdf>
- [135] J. Y. Zhu, J. Xu and V. O. K. Li, "A Four-Layer Architecture for Online and Historical Big Data Analytics," 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), Auckland, 2016, pp. 634-639.
- [136] V.Fernoagă, GA Stelea, C Gavrila, F Sandu, "Intelligent Education Assistant Powered by Chatbots", The International Scientific Conference eLearning and Software for Education, Volume 2, pp. 376-383, Carol I" National Defence University, DOI: 10.12753/2066-026X-18-122, 2018L. Xiao-Hong, "Research and Development of Web of Things System Based on Rest Architecture," 2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications, Hunan, 2014, pp. 744-747.
- [137] C. Myers, Responsive Web Design Patterns, Packt Publishing, 2015, ISBN-13: 978-1785889981
- [138] J. Pamental, Responsive Typography: Using Type Well on the Web, O'Reilly Media, 2014, ISBN-13: 978-1491907092
- [139] A. Mesbah, A. van Deursen, S. Lenselink, Crawling Ajax-Based Web Applications through Dynamic Analysis of User Interface State Changes, ACM Transactions on the Web (TWEB) - Volume 6 Issue 1, March 2012 Article No. 3, ACM New York, NY, USA.
- [140] L. Deboosere, B. Vankeirsbilck, P. Simoens, F. De Turck, B. Dhoedt, P. Demeester: Cloud-Based Desktop Services for Thin Clients. In: IEEE Internet Computing, Vol. 16, No. 6, pp. 60-67, 2012.
- [141] L. Deboosere, P. Simoens, J. De Wachter, et al: Grid design for mobile thin client computing. In: Future Generation Computer Systems, Volume 27, Issue 6, 2011, pp. 681-693, ISSN 0167-739X,
- [142] W.Jiang, M.Zhang, B.Zhou, et al: Responsive web design mode and application, 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA), Ottawa, ON, 2014, pp. 1303-1306.
- [143] G. Richards, S. Lebresne, B. Burg, J. Vitek: An analysis of the dynamic behavior of JavaScript programs. In: Proceedings of the 31st ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '10). ACM, New York, NY, USA, 2010. pp. 1-12.
- [144] E. Andreasen, A. Moller: Determinacy in Static Analysis for jQuery. In: Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications - OOPSLA '14, Portland, Oregon, USA. 2014, pp. 17-31.
- [145] GA Stelea, V.Fernoaga, C Gavrila, D Robu, "Web-Service Based Thin Client for Tele-Measurement", The International Scientific Conference eLearning and Software for Education, Volume 2, pp. 128-134, Carol I" National Defence University; DOI: 10.12753/2066-026X-18-088, 2018C. Zhao, L. Zhao, H. Wang: A spreadsheet system based on data semantic object. In: 2010 2nd IEEE International Conference on Information Management and Engineering, Chengdu, 2010, pp. 407-411.

- [146] S. Adam, U.P. Schultz: Towards Tool Support for Spreadsheet-based Domain-specific Languages. In: SIGPLAN Not. Journal, New York, NY, USA, Vol. 51, No. 3, March 2016 pp. 95-98.
- [147] S.J. Qin: Process data analytics in the era of big data. In: AIChE Journal, Volume 60, No. 9, 2014, pp. 3092-3100, ISSN: 1547-5905.
- [148] W.Wang et al - "Survey on the Communication Architectures in Smart Grid" - The International Journal of Computer and Telecommunications Networking, Vol.55, Issue 15, Oct.2011, pp 3604-3629, Elsevier, ISSN 1389-1286
- [149] Y.Yan et al - "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges" - IEEE Communications Surveys & Tutorials, Vol.15, Issue 1, Feb.2012, pp 5-20, ISSN 1553-877X
- [150] K.M.Rabie, E.Alsusae - "On Improving Communication Robustness in PLC Systems for More Reliable Smart Grid Applications" - IEEE Transactions on Smart Grid, Vol.6, Issue 6, June 2015, pp 2746-2756, ISSN 1949-3053
- [151] C.Bujdei, S.A.Moraru, "A low cost framework designed for monitoring applications, based on Wireless Sensor Networks", 13th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), Bran, 2012
- [152] M.C.Luculescu, S.Zamfira, L.Cristea, "WiSeIn: Wireless Sensor Network Used for Data Acquisition from Indoor Locations", Mechanisms and Machine Science, Vol.18, 2014, pp 391 -399 – Springer
- [153] J. Momoh, Smart Grid : Fundamentals of Design and Analysis, Hoboken, New Jersey : John Wiley & Sons, Inc., 2012, pp. 1-15.
- [154] Y. Iwase, A. Takigawa, "Extreme" Testing of WebRTC Applications WebRTC Meetup Japan, Yokohama, Japan, November 2015, [online] Available: <https://www.slideshare.net/iwashi86/extreme-testing-of-webrtc-applications>.
- [155] S. Taheri, L.A. Beni, A.V. Veidenbaum, A. Nicolau, R. Cammarota, J. Qiu, Q. Lu, M.R. Haghghat, "WebRTC Bench: A Benchmark for Performance Assessment of WebRTC Implementations", IEEE Symp. on Embedded Systems For Real-time Multimedia (ESTIMedia), Oct. 2015.
- [156] Q. Liu, J. Hu, L. Zhai, J. Huang, "Research and application of response web design for cross-platform and multi-terminal devices--Taking the migrant workers skills training and integrated services platform for example", Modern Educational Technology, no. 2, pp. 107-133, 2015.
- [157] Gayeon Kim, Sumin Kim, Eunryung Hyun, "A Study on the Evaluation of the UI-Based Usability of Different Sorts of Responsive Web Interfaces", Journal of Korea Design Knowledge, vol. 33, pp. 181-190, 2015.
- [158] T. Daisy Premila Bai, S. Albert Rabara, "Design and Development of Integrated Secured and Intelligent Architecture for Internet of Things and Cloud Computing", Future Internet of Things and Cloud (FiCloud) 2015 3rd International Conference on, pp. 817-822, 2015.
- [159] Hong-Linh Truong, Nanjangud C. Narendra, Kwei-Jay Lin, "Notes on ensembles of IoT network functions and clouds for service-oriented computing and applications", Service Oriented Computing and Applications, 2018.
- [160] Burak Kantarci, Kevin G. Carr, Connor D. Pearsall, "SONATA:", International Journal of Distributed Systems and Technologies, vol. 7, pp. 59, 2016.
- [161] Marino Linaje, Adolfo Lozano-Tello, Miguel A. Perez-Toledano, Juan Carlos Preciado, Roberto Rodriguez-Echeverria, Fernando Sanchez-Figueroa, Providing RIA user interfaces with accessibility properties, Journal of Symbolic Computation, Volume 46, Issue 2, 2011, Pages 207-217, ISSN 0747-7171, <https://doi.org/10.1016/j.jsc.2010.08.008>.
- [162] Giulio Mori (2012). Web Accessibility and Collaboration to Support Learning for Blind People (Doctoral dissertation). Retrieved from SISTEMA ETD - Banca dati delle tesi e dissertazioni accademiche elettroniche. (URN etd-04242012-121149)
- [163] Riley-Huff, Debra A. Library Technology Reports; Chicago Vol. 48, Iss. 7, (Oct 2012): 29-35. S
- [164] Shadi Abou-Zahra, Judy Brewer, and Shawn Lawton Henry. 2013. Essential components of mobile web accessibility. In Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility (W4A '13). ACM, New York, NY, USA, 4 pages. [dx.doi.org/10.1145/2461121.2461138](https://doi.org/10.1145/2461121.2461138)
- [165] Xabier Valencia, Myriam Arrue, J. Eduardo Pérez, and Julio Abascal. 2013. User individuality management in websites based on WAI-ARIA annotations and ontologies. In Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility (W4A '13). ACM, New York, NY, USA, , Article 29, 10 pages. DOI=<http://dx.doi.org/10.1145/2461121.2461128> S
- [166] P. Smutný, "Mobile development tools and cross-platform solutions," Proceedings of the 13th International Carpathian Control Conference (ICCC), High Tatras, 2012, pp. 653-656. doi: 10.1109/CarpathianCC.2012.6228727
- [167] M. Pereira, J.A. Martins, aRDF: A plugin to expose RDFa semantic information using Grails, 6th Euro American Conference on Telematics and Information Systems (EATIS), Valencia, 2012, pp 1-8.
- [168] A. Taivalsaari, T. Mikkonen, K. Systä, "Liquid software manifesto: The era of multiple device ownership and its implications for software architecture", Proceedings of the 38th Annual IEEE International Computers Software and Applications Conference, Jul. 2014.
- [169] G. Tummarello, P. Mika, Web Semantics in the Clouds, IEEE Intelligent Systems, vol. 23, September/October 2008, pp 82-87.
- [170] C. Franke, S. Morin, A. Chebotko, J. Abraham, P. Brazier, Distributed Semantic Web Data Management in HBase and MySQL Cluster, Cloud Computing (CLOUD), 2011 IEEE International Conference, pp 36 - 43.

- [171] Joorabchi, M.E., Mesbah, A., Kruchten, P.: Real Challenges in Mobile App Development. ACM / IEEE International Symposium on Empirical Software Engineering and Measurement. 2013;:2851–2864.
- [172] R. Fielding, REST: Architectural Styles and the Design of Network-based Software Architectures, University of California, Irvine, Doctoral dissertation, 2000.
- [173] B. Steele, J. Chandler, S. Reddy, Real-time Analytics, Algorithms for Data Science. (2016) Springer, Cham.
- [174] N. Enache-David, L. Sangeorzan, G.A. Stelea, "Data Analytics – Between Theory and Practice", October 2017 DOI: 10.19062/1842-9238.2017.15.2.11, Review of the Air Force Academy No. 2. Pp. 85-92, Henri Coanda Air Force Academy
- [175] Kim, J., Baratto, R.A., Nieh, J.: pTHINC: a thin-client architecture for mobile wireless web. In Proceedings of the 15th international conference on World Wide Web. New York, USA, 143-152.
- [176] D. Duggan, Service-Oriented Architecture, Enterprise Software Architecture and Design: Entities, Services, and Resources, John Wiley & Sons, 2012, pp 207-216.
- [177] Hakola, S.; Chen, T.; Lehtomaki, J.; Koskela, T. Device-To-Device (D2D) Communication in Cellular Network-Performance Analysis of Optimum and Practical Communication Mode Selection. In Proceedings of the IEEE Wireless Communication and Networking Conference, Sydney, NSW, Australia, 18–21 April 2010; pp. 21–26.
- [178] Su Yeon Kim, Sang Hoon Lee, Hyun Seok Hwang, "A Study of the Factors Affecting Adoption of a Smartphone", *Entru Journal of Information Technology*, vol. 2, 2011.
- [179] Doppler, K.; Yu, C.H.; Ribeiro, C.; Janis, P. Mode Selection for Device-To-Device Communication Underlying an LTE-Advanced Network. In Proceedings of the IEEE Wireless Communication and Networking Conference, Sydney, NSW, Australia, 18–21 April 2010; pp. 21–26.
- [180] Hyoung Yong Oh, Byoung Won Min, Yong Sun Oh, "The application of One Page Design and Responsive Web Design for Personalization Service on the Mobile Environment", *The Korea Contents Association 2012 Spring General Scientific Congress*, vol. 18, pp. 17, 2012.
- [181] Oluwatosin, H.S.: Client-server model. *IOSR Journal of Computer Engineering*. 2014;16(1):67–71. p-ISSN: 2278-8727.
- [182] Tian, Y., Song, B., Huh, E.: Towards the Development of Personal Cloud Computing for Mobile Thin-Clients, *International Conference on Information Science and Applications*, Jeju Island, 2011, pp.1-5.
- [183] G. L. Bernstein, S. Klemmer, "Towards responsive retargeting of existing websites", *Proceedings of the adjunct publication of the 27th annual ACM symposium on User interface software and technology*, pp. 119-120, 2014.
- [184] L. Sangeorzan, N. Enache-David, G.A Stelea, „AMP versus Bootstrap” 32nd IBIMA Conference: 15-16 November 2018, Seville, Spain Volume 32, ISBN: 978-0-9998551-1-9
- [185] M. Fiedler, T. Hossfeld and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," in *IEEE Network*, vol. 24, no. 2, pp. 36-41, March-April 2010.
- [186] Li, Y.; Wang, Z.; Jin, D.; Chen, S. Optimal Mobile Content Downloading in Device-to-Device Communication Underlying Cellular Networks. *IEEE Trans. Wirel. Commun.* 2014, 13, 3596–3608.
- [187] Min, H.; Seo, W.; Lee, J.; Park, S.; Hong, D. Reliability Improvement Using Receive Mode Selection in the Device-to-Device Uplink Period Underlying Cellular Networks. *IEEE Trans. Wirel. Commun.* 2010, 10, 413–418.
- [188] Yu, C.H.; Doppler, K.; Ribeiro, C.B.; Tirkkonen, O. Resource sharing optimization for device-to-device communication underlying cellular networks. *IEEE Trans. Wirel. commun.* 2011, 10, 2752–2763.
- [189] Min, H.; Lee, J.; Park, S.; Hong, D. Capacity enhancement using an interference limited area for device-to-device uplink underlying cellular networks. *IEEE Trans. Commun.* 2011, 10, 3995–4000.
- [190] V. P. Fernoaga, G.A. Stelea, A. Balan and F. Sandu, "OCR-based Solution for The Integration of Legacy And-Or Non-Electric Counters in Cloud Smart Grids," 2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME), Iasi, 2018, pp. 398-403. doi: 10.1109/SIITME.2018.8599200.
- [191] Y. Tong, J. Peng, "Research on the refactoring of benchmark course website based on media queries", *J. Software Guide*, vol. 20, pp. 111-114, 2013.
- [192] Shiri Azenkot and Emily Fortuna. 2010. Improving public transit usability for blind and deaf-blind people by connecting a braille display to a smartphone. In Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '10). ACM, New York, NY, USA, 317-318. DOI=<http://dx.doi.org/10.1145/1878803.1878890>
- [193] Andy Brown and Simon Harper. 2013. Dynamic injection of WAI-ARIA into web content. In Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility (W4A '13). ACM, New York, NY, USA, Article 14, 4 pages. DOI=<http://dx.doi.org/10.1145/2461121.2461141>
- [194] Ravi Kuber, Amanda Hastings, Matthew Tretter, and Dónal Fitzpatrick, "Determining the Accessibility of Mobile Screen Readers for Blind Users" *Imaging and Signal Processing in Health Care and Technology / 772: Human-Computer Interaction / 773: Communication, Internet and Information Technology – 2012*, DOI: 10.2316/P.2012.772-003
- [195] Ruadhan O'Donoghue, AMP: Building Accelerated Mobile Pages: Create lightning-fast mobile pages, Packt Publishing - October 2017, ISBN 139781786467317
- [196] Tanvi Dilip Challirwar, Internet of Things, *International Journal of Computer Science and Information Technology Research*, Vol. 7, Issue 2, pp: (7-12), Month: April - June 2019, ISSN 2348-1196 (print), ISSN 2348-120X (online)

Rezumat

Scopul tezei de doctorat „*Contribuții la Managementul prin Servicii Web în Internetul Obiectelor*” este de a propune și implementa noi metode pentru îmbunătățirea managementului și interoperabilității în contextul Internetului obiectelor prin *extinderea serviciilor web* de la nivelul OSI *prezentare* către nivelele inferioare. Am extins conceptul *repartizării sarcinii computaționale* Cloud computing / Edge computing de optimizare a localizării proceselor – centralizat / descentralizat. Am expus *capabilități instrumentale prin servicii web accesibile de pe “Thin Clients”*. Gradul înalt de inter-operabilitate este dat de accesul și execuția într-un navigator web. Am implementat *managementul avansat al resurselor pentru “Thin Clients” folosind “capabilități offline progresive”*. Prin această contribuție am păstrat avantajele metodelor de management orientate pe conexiune și pe specificul informației adăugându-le capabilitatea de a funcționa și offline în sensul evitării pierderilor de date / evitării pierderii sesiunii de lucru – atunci când *conexiunea este intermitentă sau lipsește pe durate mai mari*. Am definit și propus o *metodologie de acces la resurse în Internetul obiectelor* orientată pe management telematic. Am urmărit controlul unificat, compatibilitatea inter-dispozitiv și îmbunătățirea securității prin minimizarea dependențelor de terți. În acest scop am luat în considerare generalizarea conceptului de “apel” (ca “apel la resurse”, om sau mașină) incluzând descoperirea acestora (având la bază expunerea capabilităților) – printr-un ansamblu de servicii care se pot extinde / se pot scala pentru managementul personalizat al eliberării resurselor angajate. Am creat un *sistem de dezvoltare pentru aplicații web și “clienți subțiri” (Thin Clients)* menit să asigure un flux de lucru adecvat și un management orientat pe conexiune și/sau pe specificul informației pentru utilizarea mai eficientă a resurselor. În contextul stratificării IoT, al controlului bazat pe evenimente și al orientării pe servicii pentru mediul distribuit și Cloud, dar și al integrării managementului semantic cu statisticile avansate de timp real am dezvoltat un sistem de *“analiză conținutului în timp real folosind servicii web, metadate semantice și procesare Cloud”*, cu scopul de a obține performanțe îmbunătățite în vizualizarea datelor și creșterea competitivității analitice, ceea ce ar putea deschide calea către conceptul de raționament în timp real. Am integrat două demonstratoare de aplicabilitate a *“semanticii web și optimizării metadatelor în comutația fluxurilor de date pentru persoane cu impedimente de vedere și pentru cei cu acces limitat la conexiunile Internet”*. Controlul pe baza de atribute fiind implementat în urmărirea comportamentală prin metadate și conținut semantic, provocarea privind managementul resurselor în IoT a fost preluată de „declanșatoare” (triggers) care gestionează consumul de resurse. Am elaborat o serie de soluții originale privind *“Tele-măsurarea cu instrumentație virtuală utilizând servicii web”*, prin reprezentarea unificată a stărilor / evenimentelor (de tranziție) a fost posibilă o integrare mai bună a gestionării resurselor într-un mediu distribuit prin simplificarea mașinilor algoritmice de stare asociate cu modelele comportamentale ale instrumentelor. Extinzând conceptul emergent de IoT la noua paradigmă SloT (“Social IoT”), pentru a *permite obiectelor să aibă propriile rețele sociale*, am contribuit la *proiectarea și dezvoltarea unui demonstrator pe platforma IoT, pentru compoziția dinamică a serviciilor*, bazate pe principiul “serviciu ca obiect” pentru a defini un model de analiză a serviciilor, studierea modelelor de abstractizare a acestora, studierea modelelor de interacțiune socială între serviciile înseși și între servicii și utilizatori, integrarea indicatorilor de calitate tipici pentru domeniul de aplicare al SloT și optimizarea regulilor de căutare a serviciilor definite.

Rezumat (English)

Abstract:

The purpose of the PhD thesis "*Contributions to Internet of Things Management through Web Services*" is to propose and implement new methods for improving management and interoperability in the context of the Internet of Things by *extending web services* from the OSI presentation level to lower levels. The concept of Cloud computing / Edge computing was expanded to optimize process localization - centralized / decentralized. I've *exposed instrumental capabilities through accessible web services through Thin Clients*. The high degree of interoperability is given by access and execution in a web browser. *Advanced resource management for "Thin Clients" using "progressive offline capabilities"* was implemented. Through this contribution, I have retained the advantages of connection-oriented management methods and information specificity by adding the ability to offline access in order to avoid data or working session loss - *when the connection is intermittent or missing for longer periods*. I have defined and proposed a methodology for *accessing resources in the Internet of Things* through telematic management. I have aimed unified control, inter-device compatibility, and improved security by minimizing third party dependencies. For this purpose, the generalization of the concept of "call" (as "call to resources", man or machine) was considered, including the discoverability (based on the capability exposition) - through a set of services that can be expanded for personalized management. I have developed a *Thin client web development system* designed to ensure a proper workflow and connection-oriented management and / or information specificity for more resource-efficient use. In the context of IoT stratification, event-based and service-oriented control for distributed and Cloud environments, and integration of semantic management with advanced real-time statistics, I have developed a *"real-time data analysis system using web services, semantic metadata and Cloud Processing"* in order to achieve improved performance in data visualization and increased analytical competitiveness, which could pave the way for the concept of real-time intelligence. I have integrated two demonstrators of the applicability of *"web semantics and metadata optimization in data flow switching for visually impaired people and those with limited access to Internet connections"*. Asset-based control being implemented in behavioral tracking through metadata and semantic content, the challenge of resource management in IoT has been taken over by "triggers" which manages the resource consumption. I have developed a series of original solutions for *"Tele-measurement with virtual instrumentation using web services"*, through the unified representation of states/events it was possible to better integrate resource management in a distributed environment by simplifying algorithmic machines status associated with behavioral patterns of instruments. Expanding the emerging concept of IoT to the new SloT ("Social IoT") paradigm to *enable objects to have their own social networks*, I have contributed to the *design and development of a demonstrator on an IoT platform for the dynamic composition of services*, to define a service analysis model, study the abstraction patterns and patterns of social interaction between services themselves and between services and users, integrating typical quality indicators for the SloT scope, and optimize search rules for defined services.

Curriculum vitae



Nume / Prenume **STELEA George-Alex**

Experiența profesională

Perioada 2013 – 2017

Funcția sau postul ocupat **Programator / Web Developer**

Numele angajatorului **S.C. Expert Online S.R.L.**

Perioada 2014 - 2015

Funcția sau postul ocupat **Formator / Expert formare profesională
Cursuri de calificare Web Design, S.S.M și Comunicare**

Numele angajatorului **Proiect POSDRU/164/2.3/S/137770 „Calificarea ta - un viitor mai sigur!”**

Perioada 2009 - 2013

Funcția sau postul ocupat **Designer Web / Operator introducere, validare și prelucrare date**

Numele angajatorului **S.C. Euro Webmaster S.R.L-D**

Educație și formare

Perioada 2016 – Present

Calificarea/diploma obținută **Doctorand**

Programul de studii **Inginerie Electronică, Telecomunicații și Tehnologii Informaționale**

Numele și tipul instituției de învățământ **Universitatea "Transilvania" din Brașov,
Facultatea de Inginerie Electrică și Știința Calculatoarelor**

Perioada 2014 - 2016

Calificarea/diploma obținută **Diplomă de master**

Programul de studii **Aplicații pentru telefonul mobil și tehnologii internet în E-Business,
cu predare în limba germană**

Numele și tipul instituției de învățământ **Universitatea "Transilvania" din Brașov,
Facultatea de Matematică și Informatică**

Perioada	2014
Calificarea/diploma obținută	Designer Pagini Web - Cod COR: 216613
Numele și tipul instituției de învățământ/furnizorului de formare	Camera de Comerț și Industrie Brașov
Perioada	2014
Calificarea/diploma obținută	Cisco Networking - CCNA1
Numele și tipul instituției de învățământ/furnizorului de formare	Academia Tech Adviser Brașov
Perioada	2013
Calificarea/diploma obținută	Inspector în domeniul S.S.M. - Cod COR: 325723
Numele și tipul instituției de învățământ/furnizorului de formare	S.C. Prosano S.R.L.
Perioada	2012
Calificarea/diploma obținută	Manager de proiect - Cod COR: 241919
Numele și tipul instituției de învățământ/furnizorului de formare	F.M.T. Brașov
Perioada	2011
Calificarea/diploma obținută	Programare WEB (PHP/MYSQL) - Cod COR: 411303
Numele și tipul instituției de învățământ/furnizorului de formare	Școala Avantaj Consulting București
Perioada	2011
Calificarea/diploma obținută	Formator de formatori - Cod COR: 241207
Numele și tipul instituției de învățământ/furnizorului de formare	S.C. Pro Expert S.R.L.
Perioada	2011
Calificarea/diploma obținută	Tehnică Avansată HTML și Animație - Cod COR: 411303
Numele și tipul instituției de învățământ/furnizorului de formare	Școala Avantaj Consulting București

Perioada 2006 - 2009
 Calificarea/diploma obținută **Diplomă de licență**
 Programul de studii **Științe Economice și Management**
 Numele și tipul instituției de învățământ/furnizorului de formare **Universitatea "Spiru Haret" din București, Facultatea de Management Brașov**

Perioada 2009
 Calificarea/diploma obținută **Limba Engleză - Nivel II**
 Numele și tipul instituției de învățământ/furnizorului de formare **Școala Hilda Brașov**

Perioada 2008
 Calificarea/diploma obținută **Inspector Resurse Umane - Cod COR: 342304**
 Numele și tipul instituției de învățământ / furnizorului de formare **Camera de Comerț și Industrie Brașov**

Perioada 2001 - 2005
 Calificarea/diploma obținută **Diplomă de Bacalaureat**
 Numele și tipul instituției de învățământ **Colegiul Național "Andrei Șaguna" Brașov, Profilul Real Specializarea Matematică-Fizică - Științe ale Naturii**

Aptitudini și competențe personale

Limba maternă Română

Limbi străine cunoscute

Autoevaluare
 Nivel european (*)

		Înțelegere				Vorbire			Scriere	
		Ascultare		Citire		Participare la conversație		Discurs oral		
Engleză	C1	Utilizator experimentat	C1	Utilizator experimentat	C1	Utilizator experimentat	C1	Utilizator experimentat	B2	Utilizator independent
Germană	C1	Utilizator experimentat	B2	Utilizator independent	B2	Utilizator independent	B2	Utilizator independent	B1	Utilizator independent
Italiană	C1	Utilizator experimentat	C1	Utilizator experimentat	C1	Utilizator experimentat	C1	Utilizator experimentat	B2	Utilizator independent

(*) Nivelul cadrului european comun de referință pentru limbi

Curriculum vitae - English



Name **STELEA George-Alex**

Professional experience

2013 – 2017
Occupation or position held **Developer / Web Developer**
Employer Name **S.C. Expert Online S.R.L.**

2014 - 2015
Occupation or position held **Trainer / Expert trainer**
Qualification courses: Web Design and Online Communication
Employer Name **POSDRU/164/2.3/S/137770 Project „Calificarea ta - un viitor mai sigur!”**

2009 - 2013
Occupation or position held **Web Designer / Data processing operator**
Employer Name **S.C. Euro Webmaster S.R.L-D**

Education and Training

2016 – Present
Qualification/diploma **PhD Student**
Study program **Electronic Engineering, Telecommunications and Information Technologies**
Name and type of education institution **"Transilvania" University of Brasov, Faculty of Electrical Engineering and Computer Science**

2014 - 2016
Qualification/diploma **Master's degree**
Study program **Applications for mobile phone and Internet technologies in E-Business**
Name and type of education institution **"Transilvania" University of Brasov, Faculty of Mathematics and Informatics**

	2014
Qualification/diploma	Web Design – COR Code: 216613
Name and type of education institution	Chamber of Commerce and Industry Braşov
	2014
Qualification/diploma	Cisco Networking - CCNA1
Name and type of education institution	Tech Adviser Academy Braşov
	2012
Qualification/diploma	Project manager – COR Code: 241919
Name and type of education institution	F.M.T. Braşov
	2011
Qualification/diploma	Web programming (PHP/MYSQL) – COR Code: 411303
Name and type of education institution	Avantaj Consulting School Bucharest
	2011
Qualification/diploma	Advanced HTML and Animation Technique - COR Code: 411303
Name and type of education institution / training provider	Avantaj Consulting School Bucharest
	2006 - 2009
Qualification/diploma	Bachelor's degree - Management and Economics
Name and type of education institution	"Spiru Haret" University of Bucharest, Faculty of Management in Brasov
	2001 - 2005
Qualification/diploma	High school diploma
Name and type of education institution / training provider	"Andrei Şaguna" National College Brasov Mathematics-Physics - Natural Sciences