

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

DEPARTAMENTUL CALCULATOARE

Tehnici de reprezentare în Realitate Augmentată a datelor senzoriale stocate în Cloud de pe dispozitive IoT

Lucrare de disertatie

Absolvent

Virgil Emilian ANDREIES

Conducător

Prof. Dr. Ing. Dorian GORGAN

Septembrie 2017



DECAN FACULTATE

Prof. dr. ing. Liviu MICLEA

DIRECTOR DEPARTAMENT

Prof. dr. ing. Rodica POTOLEA

Tehnici de reprezentare în Realitate Augmentată a datelor senzoriale stocate în Cloud de pe dispozitive IoT

Lucrare de disertație

1. Absolvent: Virgil Emilian ANDREIES
2. Conducător: Prof. Dr. Ing. Dorian GORGAN
3. Conținutul lucrării: Introducere, Obiective și specificații, Studiu bibliografic, Fundamente Teoretice, Prezentarea contribuțiilor autorului: Prototipul AR-Cloud-IoT, Rezultate teoretice și experimentale, Configurarea prototipului și manualul de utilizare, Concluzii, Bibliografie, Anexe, CD.
4. Locul documentării: UTCN, Cluj-Napoca
5. Consultanți: Prof. Dr. Ing. Dorian Gorgan
6. Data emiterii temei:
7. Data predării:

Semnătură Conducător
Prof. Dr. Ing. Dorian GORGAN

Septembrie 2017

Semnătură Absolvent
Virgil Emilian ANDREIES



Declarație pe proprie răspundere privind autenticitatea lucrării de disertație

Subsemnatul *Virgil Emilian ANDREIES*, legitimat cu *CI* seria *CJ* numărul *028033*, *CNP 1900131060057*, autorul lucrării *Tehnici de reprezentare în Realitate Augmentată a datelor senzoriale stocate în Cloud de pe dispozitive IoT* elaborată în vederea susținerii examenului de finalizare a studiilor de masterat la Facultatea de Automatică și Calculatoare, Departamentul Calculatoare, Specializarea *Inteligentă și Viziune Artificială* din cadrul Universității Tehnice din Cluj-Napoca, sesiunea *Septembrie* a anului universitar *2016/2017*, declar pe proprie răspundere, că această lucrare este rezultatul propriei mele activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute din surse care au fost citate în textul lucrării și în bibliografie.

Declar că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență sau disertație.

În cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile administrative, respectiv, *anularea examenului de disertație*.

Cluj-Napoca
data

Semnătură
Absolvent

Rezumat

Realitatea augmentată permite vizualizarea directă sau indirectă, în timp real, a unui mediu ce reprezintă lumea reală, care a fost îmbunătățit prin adăugarea de informație generată pe calculator. De cele mai multe ori această informație este reprezentată sub forma unor obiecte virtuale 3D. Aceste obiecte virtuale sunt suprapuse peste imaginea care corespunde lumii reale. Realitatea augmentată deopotrivă îmbunătățește percepția utilizatorului asupra lumii reale și în același timp îl ajută în interacțiunea cu aceasta. Studiul de față urmărește să integreze perspectiva realității augmentate cu paradigmile Cloud și IoT. Mai exact, soluția prezintă o arhitectură experimentală care combină, conectează și explorează domeniile Internet-Of-Things (IoT), Bazele de Date NoSQL din Cloud și tehnici de Realitate Augmentată, pentru a produce date relevante și coerente, care pot fi vizualizate prin intermediul dispozitivelor mobile. În cursul căutării unei soluții la această abordare sinergetică, au fost studiate peste 50 de articole de specialitate din domeniile menționate. Rezultatul acestor studii constă într-o sinteză referitoare la concepțele și fundamentele teoretice alături de studiul diverselor tehnologii utilizabile. Apoi, s-a dezvoltat și experimentat o arhitectură descompusă în 3 mari nivele: IoT, Cloud și AR. În continuare, a fost esențială evaluarea și validarea experimentală a funcționalității și performanțelor soluției propuse. Acestea s-au realizat prin raportare la performanță, scalabilitate, utilizabilitate, și portabilitate.

Cuprins

Listă de tabele	iii
Listă de figuri	v
1 Introducere	1
1.1 Context	1
1.2 Motivatie	2
1.3 Structura lucrării	2
2 Obiective și specificații	5
2.1 Domenii de studiu	5
2.1.1 IoT (Internet-of-Things)	5
2.1.2 Infrastructuri și stocare în cloud	6
2.1.3 Realitatea augmentată	6
2.2 Obiective	7
3 Studiu bibliografic	9
3.1 Internetul lucrurilor	9
3.1.1 Generalități, definiții și studii conceptuale	9
3.1.2 Rețele IoT Wireless, Senzori, Obiecte web Inteligente	11
3.1.3 IoT și Cloud	13
3.2 Infrastructuri Cloud	13
3.2.1 Generalități, definiții și studii conceptuale	13
3.2.2 Stocare și baze de date în Cloud	15
3.3 Realitate augmentată	17
3.3.1 Generalități, definiții și concepte în AR	17
3.3.2 Direcții de studiu și tehnologii	19
4 Fundamente Teoretice	21
4.1 Internet-of-Things	21
4.1.1 Modele de comunicare in Internet of Things	22
4.2 Infrastructura Cloud	26
4.2.1 Paradigma NoSQL	29

4.3	Realitatea Augmentată	30
4.3.1	Realitatea augmentată simplă	32
4.3.2	Detectia bazata pe marker	33
5	Prezentarea contributiilor autorului: Prototipul AR-Cloud-IoT	37
5.1	Proiectarea si modelarea componentelor arhitecturale	40
5.1.1	Faza I: Achizitia si transmisia datelor	43
5.1.2	Stocarea in Cloud	48
5.1.3	Faza III: Aplicatia mobilă de Realitate Augmentată	50
5.2	Tehnologiile implementate	56
6	Rezultate teoretice si experimentale	59
6.1	Specificatii Non-Functionale	59
6.1.1	Performanta	59
6.1.2	Scalabilitate	61
6.1.3	Utilizabilitate	62
6.1.4	Portabilitate	63
6.2	Scenariu 1 - Casă automatizată	63
6.3	Scenariu 2 - Seră automatizată	63
6.4	Scenariu 3 - Centru de date	64
7	Configurarea prototipului si manualul de utilizare	67
7.1	Instructiuni de instalare si configurare	67
7.2	Instructiuni de utilizare	71
8	Concluzii	75
8.1	Rezumatul realizarilor	75
8.2	Îmbunătățiri viitoare	76
Bibliografie		77
A	Fragmente cod sursă	83
A.1	Aplicatia mobilă - C#	83
A.2	Achizitie valori senzor - Python	88
A.3	Transmisie valori senzori - JavaScript/NodeJs	90

Listă de tabele

2.1	Obiective secundare	8
8.1	Contribuții personale	76

Listă de figuri

4.1	Modelul device-to-device conform [1]	22
4.2	Modelul device-to-cloud conform [1]	23
4.3	Modelul device-to-gate conform [1]	24
4.4	Modelul back-end data sharing model conform [1]	25
4.5	IoT ca rețea de rețele conform [2]	25
4.6	Arhitectura unui cloud conform [3]	27
4.7	Exemplu de modele Cloud: Public, Privat, Hibrid preluat din [4]	29
4.8	Model de head-mounted-display conform [5]	31
4.9	Continuumul Reality-Virtuality creat de Milgram din [6]	31
4.10	Exemplu de configurare a unui sistem AR din [6]	32
4.11	Organograma unui sistem simplu AR, preluat din [6]	32
4.12	Imaginea din stânga: model augmentat amplasat asupra unui plan de clădire. Imaginea din dreapta: exemplu de marker. Din [6]	33
4.13	Figura din stânga: imagine originală; Figura din dreapta: imaginea după tehnica de "adaptive threshold". Din [6]	35
4.14	Exemplu de detecție a muchiilor. Figura din stânga: contururi detectate; Figura din dreapta: muchiile care rămân după aplicarea testului celor 4 colțuri pentru pătrate. Din [6]"	35
4.15	Exemplu de potrivire a muchiilor. Figura din stânga: potrivirea muchiilor în mediu fără distorsiuni; Figura din dreapta: detecția muchiilor peste imaginea originală. Din [6]	36
4.16	Un cub augmentat deasupra marker-ului detectat. Sistemul de coordonate (X,Y,Z) este randat cu (roșu, verde, albastru). Din [6]	36
5.1	Diagrama bloc de arhitectură	38
5.2	Diagrama Use Case	39
5.3	Diagrama de lansare	40
5.4	Diagrama de componente	42
5.5	Componența de colectare a datelor senzoriale	43
5.6	Modelul de date.	44
5.7	Organograma achiziției de date.	45
5.8	Web-server/Serviciul REST	46

5.9	Organograma transmisiei de date.	47
5.10	Achiziția și transmisia datelor senzoriale.	48
5.11	Exemplu DynamoDB cu chei de partităionare și sortare.	49
5.12	Stocarea datelor provenite de pe dispozitivul IoT.	49
5.13	Achiziționarea datelor din cloud pe aplicația mobilă	50
5.14	Structura aplicației mobilă	51
5.15	Componentele aplicației	52
5.16	Diagrama de clase	53
5.17	Diagrama de clase	54
5.18	Diagrama de clase	55
5.19	Stiva de tehnologii	56
6.1	Stocarea datelor în contextul consolei locale	60
6.2	Stocarea datelor în contextul consolei AWS	61
6.3	Achiziția datelor în contextul aplicației mobile	62
6.4	Utilizabilitate	62
6.5	Temperatura pe durata unei săptămâni	63
6.6	Umiditatea pe durata unei săptămâni	64
6.7	Presiunea pe durata unei săptămâni	65
7.1	Aplicația pe Android Motorola Moto X	71
7.2	Aplicația înainte de scanarea țintei	72
7.3	Aplicația după scanarea țintei	72
7.4	Obiectul cu datele 3D mapate pe țintă	73
7.5	Obiectul rotit cu datele 3D mapate pe țintă	73
7.6	Reprezentare în modul portret	74
7.7	Captura ecranului cu graficul virtual	74
7.8	Închiderea aplicației	74

Capitolul 1

Introducere

Subiecte precum Internet of Things, Infrastructuri Cloud și Realitatea augmentată reprezintă puncte de interes maxim în cercurile științifice și academice. În ultimii ani, aceste domenii au avut un impact major în modul în care sunt gestionate afacerile, media și chiar actualitatea socială. Fără îndoială, aceste beneficii au rezultat ca urmare a progreselor tehnologice în ceea ce privește soluțiile, tehniciile și librăriile asociate cu cele 3 domenii de studiu.

1.1 Context

Odată cu explozia tehnologică a dispozitivelor mobile (smartphone-uri și tablete), alături de vehiculele inteligente, home-automation, paradigma smart cities și smart grids, a apărut și nevoia de stocare a datelor în mod rapid, intelligent și fiabil. De asemenea, accesul la aceste medii de stocare a datelor se orientează spre servicii cu formate "light-weight" care pot fi consumate ușor și rapid. În ceea ce privește numărul total de dispozitive conectate la internet, Cisco IBSG - [2] a calculat că 12.5 miliarde de dispozitive au fost conectate la Internet în 2010. Acest lucru înseamnă că raportul de dispozitive conectate la Internet per persoană este supra-unitar(1.84). În continuare la acest lucru, CISCO a estimat că până în anul 2020 numărul de dispozitive conectate la Internet va crește până la 50 de miliarde. O consecință care reiese din această predicție este nevoia de noi modalități de stocare și procesare a datelor.

În momentul actual, cloud computing constituie un domeniu mult mai dezvoltat și matur decât celelalte menționate. În momentul de față, infrastructurile cloud oferă soluții eficiente care implementează gestiunea serviciilor dar și aplicații care procesează și exploatează datele produse de aceste servicii. Pe de altă parte, soluțiile Cloud beneficiază de pe urma IoT extinzându-i scopul pentru a lucra cu obiecte și concepte din lumea reală într-o manieră distributivă și dinamică. Această integrare duce la livrarea de servicii noi în mod continuu, în contextul scenariilor din lumea reală.

Realitatea augmentată (sau realitatea virtuală îmbunătățită) reprezintă un domeniu

1. Introducere

relativ nou, deși numeroase studii, cercetări și descoperiri au avut loc în anii 90 și 2000. Prin AR se oferă o perspectivă și o vizualizare directă sau indirectă a unu mediu real, adăugând informație care combină obiectele reale cu cele virtuale. Aceste obiecte virtuale afișează informații pe care utilizatorul nu le poate detecta prin simțurile proprii iar aceste informații îl ajută să execute sarcini din lumea reală.

1.2 Motivație

Motivația din spatele lucrării este absența din domeniile de specialitate a unei abordări care să integreze vizualizarea oferită de realitatea augmentată a datelor senzoriale captate de pe microcontrolere și stocate în baza de date a unei infrastructuri Cloud. Această abordare sinergică aduce o contribuție importantă în integrarea AR cu alte domenii pentru a produce aplicații care să îmbunătățească și să ușureze anumite aspecte ale vieții omului.

Această teză urmărește să ofere o abordare care combină, conectează și explorează domeniile de Internet-of-Things (IoT), Bazele de date din Cloud și tehnici de Realitate Virtuală Augmentată, pentru a produce date relevante, coerente și ușor de utilizat, vizualizate prin intermediul dispozitivelor mobile. Mai exact, soluția prezentă implică achiziția datelor la intervale regulate de timp de pe diferiți senzori atașați dispozitivelor micro-controlere. Apoi, informația colectată de la senzori este transmisă în Cloud și stocată într-o bază de date NoSQL. În final, o componentă mobilă interoghează serviciul din Cloud printr-un serviciu REST și achiziționează datele senzoriale la decizia utilizatorului. Utilizatorul scanează imaginile întâi atașate senzorului iar datele corelate cu acel senzor vor fi preluate în timp real. Data este apoi augmentată prin intermediul unor tehnici de vizualizare și reprezentate pe dispozitivul utilizatorului. Aceasta va interacționa cu aceste date prin diverse reprezentări 3D (ex. chart 3D, pie 3D, bar 3D).

1.3 Structura lucrării

Lucrarea este structurată după cum urmează:

În partea de început se prezintă tematica lucrării, obiectivele și motivația tezei.

Apoi, este prezentat un studiu bibliografic unde sunt abordate peste 50 de lucrări de specialitate din cele 3 domenii mai sus menționate. Vor fi abordate atât separat cât și combinate în contextul corespunzător.

Urmează prezentarea contribuțiilor personale unde se va detalia abordarea din punct de vedere tehnic. Soluția arhitecturală va fi descrisă în detaliu, urmând o abordare "general-specific" și de jos în sus. Componentele esențiale, platformele, librăriile, pachetele și limbajele de programare vor încheia această secțiune.

În continuare, este necesar ca soluția arhitecturală să fie evaluată și testată conform unor situații experimentale. Este important să fie prezentate scenarii specifice, măsurabile, accesibile, realiste și bazate în timp. Soluția va fi raportată la specificații non-funcționale

care țin de performanță, scalabilitate, utilizabilitate și portabilitate. Teza va oferi evaluări ale soluției și experimente care să o valideze prin maparea acestelor pe scenarii cât mai realiste și plauzibile.

Nu în ultimul rând, concluziile și contribuția personală vor fi expuse la final. Îmbunătățirile viitoare sunt prezentate pentru a ilustra importanța studiului curent în contextul domeniilor de studiu.

În rezumat, abordarea curentă va prezenta o combinație sinergetică între: Internet-Of-Things, Infrastructuri Cloud și Baze de Date NoSQL în cloud și Realitatea Augmentată. Teza este structurată de o astfel de natură încât fiecare domeniu va fi studiat detaliat, însă accentul se va situa pe domeniul realității augmentate.

Capitolul 2

Obiective și specificații

2.1 Domenii de studiu

Studiul prezent reprezintă intersecția a 3 domenii distințe din știința calculatoarelor, la fel ca majoritatea cercetărilor de actualitate. După cum se menționa mai sus, se vor aborda 3 teme în vogă în aceste zile, IoT, Cloud și AR. Însă realitatea augmentată reprezintă punctul maxim de interes. În consecință, fiecare domeniu va fi studiat separat, urmând ca în final ele să fie combinate și coordonate. Cu toate acestea, este de menționat că, deși nu sunt o parte integrală din teză, câteva alte domenii din știința calculatoarelor sunt gestionate, cum ar fi: Ingineria Software, Grafică pe Calculator, Rețele de Calculatoare și Sisteme Distribuite.

2.1.1 IoT (Internet-of-Things)

Internet-of-Things este un topic în plină dezvoltare cu o deosebită importanță din punct de vedere tehnic, social și economic. Produse consumabile, bunuri durabile, autovehicule, componente și utilități industriale, senzori și obiecte de uz zilnic sunt combinate folosind conectivitatea prin Internet și componente cu capacitați analitice ce promit să schimbe modul în care lucrăm, trăim și ne relaxăm. Previziunile asupra impactului IoT asupra Internet-ului și a economiei sunt impresionante. Se prefigurează ca până în anul 2025 numărul de dispozitive interconectate prin internet să ajungă la 11 trilioane. Acest lucru va avea în mod evident un impact major asupra economiei mondiale. IoT angrenează o mulțime extrem de variată de idei complexe interconectate prin diferite perspective. Acest concept de a combina calculatoare, senzori și rețele de calculatoare în vederea monitorizării și a controlului dispozitivelor a existat de foarte mult timp. Convergența diferitor trend-uri ale pieței aduce IoT mai aproape de realitatea cotidiană de la nivelul de concept sau cercetare. Câteva dintre aceste trend-uri sunt Ubiquitous Connectivity, rețele bazate pe IP, Miniaturizare, Economii de calcul, descoperiri în Data Analytics și expansiunea infrastructurilor Cloud.

2.1.2 Infrastructuri și stocare în cloud

Chiar dacă rulăm aplicații care se bazează pe "photo-sharing" printre alte milioane de utilizatori de telefoane mobile sau suntem interesați mai degrabă de operațiuni de business, o platformă de servicii cloud oferă acces rapid la resurse flexibile și cu cost redus. Utilizând Cloud Computing, nu mai este neapărat necesar să se investească în componente hardware și să se irosească timp prețios în menenanța acestor componente. În schimb, se pot gestiona atât topul cât și dimensiunea resurselor de calcul care sunt necesare pentru susținere ideile sau a opera un departament IT. Se pot accesa exact atâtea resurse cât este nevoie și costul este în funcție de cât se utilizează. Cloud Computing oferă accesul ușor la servere, spațiu de stocare a datelor, baze de date cât și un set variat de servicii prin Internet. O platformă de servicii precum Amazon Web Services (AWS) deține și menține componentele hardware cât și infrastructura de rețea necesară pentru a rula aceste servicii de aplicații, în timp ce utilizatorul folosește ceea ce are nevoie prin intermediul unei aplicații web. În cadrul Cloud Computing, se disting 3 mari tipuri care sunt deseori denumite: *Infrastructure as a Service (IaaS)*, *Platform as a Service (PaaS)*, and *Software as a Service (SaaS)*.

2.1.3 Realitatea augmentată

Realitatea augmentată (AR) poate fi definită ca vizualizarea directă sau indirectă, în timp real, a unu mediu real în lumea reală care a fost îmbunătățit/augmentat prin adăugarea de informație generată pe calculator. AR este deopotrivă interactiv și înregistrat în mediul 3D cât și combina obiectele reale cu cele virtuale.

Realitatea augmentată are ca scop simplificarea anumitor aspecte din viața utilizatorilor adăugând informație virtuală în mediul lui înconjurător dar și în percepții indirekte ale mediului real, cum ar fi imagini live de pe camere video sau foto. AR îmbunătățește percepția și interacțiunea utilizatorului asupra lumii reale. În timp ce Realitatea Virtuală (VR) cufundă total utilizatorii într-o lume sintetică fără a avea cunoștiință despre lumea reală, tehnologiile AR augmenteză simțul realității, suprapunând obiecte virtuale și indică realitatea în timp real. Teoretic, AR se poate adresa tuturor simțurilor, augmentând miroslul, atingerea și auzul în același timp. De asemenea, în multe cazuri este folosit pentru a înlocui anumite simțuri prin substituția senzorială. De exemplu, augmentarea vederii la persoanele cu dizabilități de vedere sau augmentarea auzului pentru utilizatori cu deficiențe de auz, utilizând indicații vizuale.

Obiectele virtuale adăugate în mediul real reprezintă informația către utilizator, pe care acesta nu o poate detecta direct prin intermediul simțurilor. Informația pasată de către obiectele virtuale îl pot ajuta să îndeplinească diferite sarcini la muncă, cum ar fi ghidarea muncitorilor printre grupări de fire electrice în construcția unei aeronave afișând informația digitală prin intermediului unu dispozitiv montat pe cap. Informația poate avea, în același timp și caracter de divertisment, cum ar fi "Wikitude", "Vuforia", sau "Metaio" sau orice altă aplicație de AR. Există multe alte utilizări ale aplicațiilor AR, cum ar fi vizualizarea medicală, divertisment, advertising, menenanță și reparății, adnotări,

planificarea drumului optim al unui robot, etc.

2.2 Obiective

Obiectivul principal al acestei teze de disertație este de a proiecta, modela, implementa și testa o soluție care urmărește să reprezinte în contextul realității augmentate, datele senzoriale stocate într-un serviciu de baze de date din cloud achiziționate de pe dispozitive microcontroller. Cu alte cuvinte, data este percepță de senzorii atașați microcontroller-ului, este transmisă în cloud prin intermediul unui serviciu, iar apoi este salvată și persistată în cloud. În final, prin procesul de augmentare, datele pot fi vizualizate sub formă de model grafic tri-dimensional pe un dispozitiv mobil. În concluzie, această ultimă parte constituie contribuția personală a studiului prezent.

Pentru a realiza obiectivul principal ante-menționat, se vor îndeplini următoarele obiective secundare(suport):

2. Obiective și specificații

Nr.	Obiectiv Secundar	Abordare în teză	Cap.
1	Studiu critic al cercetărilor actuale	Peste 40 de articole și cărți au fost, studiate, organizate, identificându-se posibilitatea unei sinergii între IoT, Cloud și Realitate Augmentată	3
2	Studiul și cercetarea în IoT cu privire la soluții, tehnologii și direcții de ultimă generație	Determinarea domeniilor unde soluțiile ar avea utilitate maximă în termeni de dispozitive micromănușabile și senzori; decizie cu privire la tehnologiile fezabile în acest context.	4
3	Studiul Infrastructurilor Cloud, stocare în cloud și paradigma NoSQL	Studiu comparativ asupra infrastructurilor și serviciilor prezente; direcțiile de studiu și studiu comparativ asupra serviciilor de baze de date	4
4	Cercetarea articolelor actuale din Realitatea Augmentată în contextul paradigmelor Cloud-IoT.	Investigarea direcțiilor de studiu a problemelor, limitărilor și identificarea obiectivă a tehnologiilor disponibile și a posibilităților de folosire.	4
5	Proiectarea, modelarea și implementarea unei soluții care să cuprindă abordările, tehniciile și metodologiile studiate și să introducă realitatea augmentată ca și componentă alături de cloud și IoT.	Identificarea componentelor care vor lua parte în elaborarea unei soluții prototip și implementarea efectivă a unei soluții care achiziționează date senzoriale, le stochează în cloud iar apoi le reprezintă în AR.	5
6	Validarea soluției prin scenarii experimentale care să vizeze fiabilitatea, fezabilitatea și scalabilitatea soluției în scenarii ale lumii reale.	Maparea soluției arhitecturale pe scenarii care să evaluateze timp de acces, volume de date, valabilitatea, corectitudinea și utilizabilitatea.	6

Tabela 2.1: Obiective secundare

Capitolul 3

Studiu bibliografic

Această secțiune oferă o imagine de ansamblu asupra literaturii de specialitate în ceea ce privește Internet-Of-Things, Cloud Computing și baze de date în Cloud împreună cu Realitate Augmentată.

Literatura din domeniul IoT acesta gravitează în jurul temelor cum ar fi rețele wireless de senzori, "ubiquitous sensing", medii "smart", orașe "smart", graful lucrurilor, etc. Studiile referitoare la infrastructurile Cloud urmăresc cercetarea arhitecturilor cloud, baze de date NoSQL, securitate, intimitate, tehnici de stocare, Big Data, Analytics și monitorizare. Actualitatea în materie de cercetare din realitatea augmentată abordează următoarele teme: Realitate și medii Virtuale, medii interactive de învățare, tehnologii de dezvoltare AR, Interfețe-Utilizator, Dispozitive mobile de calcul, librărie Vuforia.

3.1 Internetul lucrurilor

Categoriile sunt clasificate în următorul mod: analiza și studii de expertiză cu privire la starea actuală, rețele wireless de calculatoare și senzori, obiecte web smart, IoT și Cloud.

3.1.1 Generalități, definiții și studii conceptuale

În [2] se prefigurează că Internet of Things (IoT), de multe ori numit Internetul Obiectelor, va schimba totul. Aceasta poate părea ca o afirmație extrem de curajoasă, dar luând în calcul impactul internetului asupra educației, comunicărilor, afacerilor, științei, formelor de guvernare și umanității în general, este mai mult decât o posibilitate. În mod evident, internet-ul este una dintre cele mai revoluționare creații din istoria omenirii. Reflectând asupra faptului că IoT reprezintă următorul pas în evoluția internetului prin unelte capabile să achiziționeze, analizeze și să distribuie date care ulterior sunt preschimbate în informație și cunoștiințe. În acest context, IoT are o importanță deosebită. Există deja numeroase proiecte care oferă posibilitatea de a ajuta oamenii cu situații financiare precare, de a îmbunătăți distribuția resurselor către cei ce au nevoie mai mare și nu în

3. Studiu bibliografic

ultimul rând, să ne ajute să înțelegem mai bine planeta, fiind mai pro-activi. Cu toate acestea, există anumite obstacole care încă încetinesc dezvoltarea domeniului, inclusiv tranzitia spre IPv6, standardizarea, dezvoltarea surselor de alimentare pentru milioane sau chiar miliarde de senzori. În orice caz, cu cât mediul afacerilor, guvernele, grupurile de standardizare și academice cooperează mai repede pentru a tece peste aceste obstacole, IoT va continua să progreseze.

Definițiile, tehnologiile suport pentru dezvoltare, modele de conectivitate și potențialul transformator sunt identificate ca și componente-cheie care constituie fundamentul explorării oportunităților și obstacolelor viitoare și conceptualizare în articolul [1]. De asemenea, 5 domenii cheie din IoT sunt identificate pentru a explora obstacolele referitoare la tehnologii. Printre cele mai importante, se regăsesc: securitate, caracterul personal, interoperabilitatea și standardele, legalitatea, reglementarea și drepturile, economiile emergente și dezvoltarea. Internetul lucrurilor implică un set complex și evolutiv de considerații tehnologice, sociale și considerante politice între diferiți "stakeholderi".

Pe de altă parte, depășirea unora dintre obstacolele tehnice și socio-economice de lansare și configurare a sistemelor IoT la scară largă în viața cotidiană, necesită o evaluare practică a soluțiilor IoT cu ajutorul unor bancuri de încercare (testbeds), cu caracter interdisciplinar, de scară largă și realistic. Acestea sunt axate spre proiectarea și lansarea mediilor experimentale care permit: evaluarea în condiții realistice, evaluarea acceptării sociale a noilor soluții IoT și cuantificarea utilizabilității serviciilor și performanței cu participarea directă a utilizatorului final. Articolul [7] are drept obiectiv identificarea specificațiilor principale pentru facilitățile experimentale de ultimă generație pentru IoT. În același timp descrie o taxonomie și un studiu cu privire la bancurile de încercare disponibile în acest moment în lume. În plus, caracteristicile principale sunt mapate pe nevoi experimentale și o propunere pentru noile generații de bancuri este expusă. De asemenea, se vorbește despre WSN (Wireless Sensor Networks), se expune faptul că există deja fundamente în acest sens, dar obstacole precum heterogenitatea, repetabilitatea, federația, concurența, mediile experimentale, mobilitatea și implicarea utilizatorului trebuie încă studiate.

GSMA (Asociația GSM) ajunge la concluzia că este de așteptat ca IoT să se răspândească foarte repede în anii ce urmează iar această convergență cu alte domenii va dezlașui o nouă dimensiune în ceea ce privește serviciile care îmbunătățesc calitatea vieții pentru consumatori, cât și productivitatea pentru întreprinderi și afaceri. În [8] acest lucru este denumit "Connected Life". Desigur că punctele de vedere sunt cele ale consumatorului și producătorului iar îmbunătățiri cum ar fi sănătatea și educația, manufactura și retail, agricultura vor impacta viața oamenilor. În acest sens, soluțiile Machine-to-Machine (M2M) deja oferă rețele wireless pentru inter-conectarea dispozitivelor prin Internet, cu intervenție umană minimă, pentru a oferi servicii care se aliniază nevoilor diferitor industriei. Fiind o evoluție a M2M, IoT reprezintă o coordonare a diferitor tipuri de mașini, dispozitive și aparate conectate prin diferite rețele din Internet.

Articolul [9] determină 3 caracteristici importante care guvernează IoT: 1) Instrumentarea obiectelor obișnuite; 2) Interconectarea terminalelor autonome; 3) Servicii inteligente universale. Viziunea pentru un sistem IoT cuprinde o împărțire a arhitecturii

în patru nivele: Nivelul Obiectelor Senzoriale, Nivelul de schimb de date, Nivelul de Integrare a Informației și Nivelul de Servicii Aplicație. Nivelul de senzori captează datele senzoriale; nivelul de date gestionează transparent transmisia de date; nivelul de integrare se ocupă de recombinarea, filtrarea și fuziunea informațiilor cu caracter nesigur iar apoi integrează această informație în cunoștiințe utilizabile; nivelul de servicii expune conținutul prin servicii spre diversi utilizatori.

În acest sens, obiectivele IoT sunt definite ca extensibilitate în ceea ce privește dispozitivele, percepția intensivă a informațiilor și servicii inteligente cât mai cuprinzătoare. După cum a fost menționat și în [9], câteva probleme științifice cheie se referă la schimbul de date între elemente eterogene de rețea la scară largă, integrarea eficientă și adaptarea interacțiunii, a informațiilor nesigure și adaptarea serviciilor în mediul dinamic al sistemului.

Whitmore, în [10], ajunge la concluzia că analiza literaturii de specialitate din IoT se axează în mare parte pe tehnologie în acest moment. Acest lucru pare evident devreme ce IoT nu a ajuns încă la o maturitate care acceptabilă care să fie concretizată prin producție la scară largă. De îndată ce tehnologia va ajunge la maturitate, cercetarea va fi nevoie să se extindă pe diferite arii de specializare: management, operații, reglementare, economie și sociologie. Revizuirea literaturii a dat naștere unor constatări importante care pot concentra eforturile de cercetare. Se afirmă că IoT nu este bine reprezentat în literatura de management iar standardele și cercetarea din IoT sunt dominate de studii în Europa și Asia. Literatura IoT este dominată de cercetarea referitoare la tehnologiile IoT.

3.1.2 Rețele IoT Wireless, Senzori, Obiecte web Inteligente

Un pas important către IoT ar fi facilitarea tehnologiilor adecvate pentru rețelele de senzori wireless bazate pe un protocol standard verificat, Protocolul de Internet (IP), pentru a sprijini rețeaua de lucruri (Internet-of-Things). Lucrarea [11] introduce o nouă abordare numită SNAIL, care include o descriere completă a arhitecturii ce integrează patru tehnologii importante - mobilitate, activare web, sincronizare de timp și securitate la arhitectura de adaptare IP primară. Arhitectura de bază încorporează protocoalele necesare realizării întregii operațiuni IP pe care standardul nu le specifică.

Studiul de expertiză din [12] identifică faptul că IoT apare ca una dintre tendințele majore care modeleză dezvoltarea tehnologiilor în sectorul ITC. În plus, se precizează că este iminentă trecerea de la un Internet utilizat pentru interconectarea dispozitivelor utilizatorilor finali la un internet folosit pentru interconectarea obiectelor fizice care comunică între ele și/sau cu omul. Se face referire la conceptul binecunoscut de "obiecte inteligente" sau "lucruri inteligeante". Din punct de vedere conceptual, se afirmă că IoT se bazează pe capacitatea obiectelor inteligente de a: 1) fi identificabile; 2) comunica și 3) interacționa. La nivel de componente, "obiectele inteligente" vor completa entitățile existente, cum ar fi hosts, terminalele, routerele etc. Aceste "obiecte inteligente" sunt identificate ca entități care: au o reprezentare fizică, au un set de funcționalități, sunt identificabile în mod unic, asociate cu cel puțin un nume sau o adresă, posedă unele capacitați computaționale de bază

3. Studiu bibliografic

și, cel mai important, pot sesiza fenomenele fizice (temperatura, lumina, nivelul radiațiilor electomagnetice) sau pot declanșa acțiuni care au un efect asupra realității fizice (actuatori). Aplicațiile și zonele de impact variază de la locuințe inteligente/clădiri inteligente, orașe inteligente, monitorizare de mediu la îngrijirea sănătății, afacerea/inventarul intelligent și gestiunea produselor, precum și securitatea și supravegherea.

Există câteva direcții de studiu care explorează noțiunile de "obiect social" în corelație directă cu Internetul obiectelor pentru a oferi o integrare cu concepte cum ar fi rețelele sociale. În [13], sunt identificate și detaliate trei etape care implică creșterea nivelului de implicare socială a obiectelor care compun Internetul obiectelor. Aceste etape implică postarea de informații despre starea lor în rețelele sociale ale oamenilor, obiectele pot interacționa la nivelul aplicației în rețelele sociale cu oamenii și alte obiecte și, în final, pot interacționa între ele pentru a construi o rețea de comunicații. În plus, 2 cazuri de utilizare au fost abordate în contextul relațiilor sociale dintre obiecte. În același timp, unele probleme de cercetare legate de subiectul "comportamentul social al obiectelor inteligente" dovedește că într-adevăr, maturitatea unor astfel de concepte se află în faza inițială.

O concluzie cu privire la industria software este tendința de a trece la tehnologii integrate orientate spre servicii. Mai ales în domeniul software-ului de afaceri, au fost dezvoltate aplicații complexe bazate pe compozitia și colaborarea între diverse servicii. Conceptul de "Internet de servicii" presupune acest lucru la scară largă, în care serviciile se află în diferite nivele ale întreprinderii (unități operaționale, rețele IT sau chiar care rulează direct pe dispozitive și mașini din cadrul companiei - on premise). În [14], este clar că noile oportunități de integrare mai bună în timp real a lumii fizice au ca rezultat nevoia de servicii diversificate și eficiente dinamic. În acest context, al SOA (Service Oriented Architectures), se propune un proces și un sistem adecvat, care permite dezvoltatorilor și proiectanților de procese de afaceri să interogheze dinamic, să selecteze și să utilizeze instanțe de execuție ale serviciilor din lumea reală (pe dispozitive fizice) și să implementeze altele noi, la cerere, în contextul aplicațiilor de afaceri din lumea reală. În plus, se afirmă că, în viitor, dispozitivele din lumea reală vor putea oferi funcționalitatea prin intermediul serviciilor web bazate pe SOAP (WS-*) sau API REST, permitând interacțiunea dinamică cu alte componente. În această direcție de gândire, abordarea prezentată nu numai că ar putea să caute eficient serviciile care rulează pe dispozitive încorporate, ci și să implementeze funcționalități lipsă la cerere.

Procesul sugerează că este posibilă extinderea domeniului de enterprise computing la lumea reală (și invers). Se recomandă utilizarea standardelor de servicii web (DPWS) și a modelelor web orientate (REST) pentru a integra cu ușurință dispozitivele fizice în sistemele informatice existente. Serviciile Web pe dispozitive pot fi utilizate pentru înregistrarea dinamică a dispozitivelor și serviciilor pe care le furnizează.

Articolul [15] urmărește o inițiativă și un proiect extrem de ambițioase, anume conectarea unei cantități mari de elemente de date disponibile, generate de IoT, către restul lumii digitale, pentru a transforma aceste date în acțiuni semnificative, care vor duce la crearea de noi capabilități, experiențe mai bogate și oportunități economice fără precedent pentru întreprinderi, persoane și țări. Cu toate acestea, furnizarea unei vizualizări integrate

pentru explorarea și interogarea unor astfel de date în timp real este extrem de provocatoare datorită caracterului "Big-Data" al datelor: volum mare, actualizare rapidă în timp real și surse de date ne-uniforme. Pentru a răspunde acestei provocări, se oferă o vizualizare integrată și live pentru surse de date eterogene IoT utilizând Linked Data, denumită Graph of Things (GoT). GoT este susținut de un pachet software scalabil și elastic pentru a face față miliardelor de înregistrări de seturi de date istorice și statice, împreună cu milioane de triplete, fiind preluate și îmbogățite pentru a conecta GoT în timp real. GoT face ca fluxul de aproximativ jumătate de milion de surse de date să poată fi interogate printr-un endpoint SPARQL și un canal de interogare continuă folosind Websockets care permite crearea unui sistem explorator live al GoT la adresa <http://graphofthings.org/> folosind doar cod HTML și Javascript.

3.1.3 IoT și Cloud

Este greu în această epocă să se abordeze subiectul IoT sau Internetul Obiectelor, Internet of Everything fără o corelație în vreun fel sau altul cu Infrastructurile Cloud în ceea ce privește stocarea, analiza, procesarea, monitorizarea etc. Paradigma Internetului Obiectelor (IoT) se bazează pe noduri inteligente și auto-configurabile interconectate într-o infrastructură de rețea dinamică și globală. Aceasta reprezintă una dintre cele mai perturbatoare tehnologii care din "ubiquitous computing". Internetul lucrurilor se referă, de obicei, la lumea reală și lucrurile mici cu capacitatea de stocare și prelucrare limitată, precum și problemele importante legate de fiabilitate, performanță, securitate și confidențialitate. Pe de altă parte, cloud computing are capacitatea aproape nelimitată de stocare și putere de procesare, reprezentând o tehnologie mai matură, cel puțin în anumite aspecte și rezolvă o parte din problemele internetului lucrurilor. În [16] se prezintă o nouă paradigmă, denumită *Cloud IoT*. În aceasta, cele 2 tehnologii coopereză pentru a atinge un obiectiv. Pe de altă parte, alte perspective precum [17] văd cele 2 tehnologii ca opuse.

În cele din urmă, [18] prezintă un model bazat pe cloud pe bază de utilizatori pentru gestionarea datelor noi în evoluție, prin interacțiunea dintre cloud-ul privat și cel public. În acest mod, nevoile utilizatorului final sunt aduse în prim plan. Permitând flexibilitatea necesară pentru a răspunde nevoilor diverse și uneori concurente ale diferitelor sectoare, se propune un framework activat de un cloud scalabil, care să ofere capacitatea de a utiliza IoT.

3.2 Infrastructuri Cloud

3.2.1 Generalități, definiții și studii conceptuale

Termenul de "cloud" pare să aibă originea în diagramele de rețea care reprezentau internetul sau diverse părți ale acestuia, ca și nori schematici. "Cloud computing" a fost inventat pentru ceea ce se întâmplă atunci când aplicațiile și serviciile sunt mutate în "cloud".

3. Studiu bibliografic

Cloud computing nu este ceva care a apărut brusc peste noapte; într-o anumită formă se poate merge în timp până într-o perioadă în care sistemele informatiche împărtășeau la comun resurse și aplicații de calcul în funcție de un anumit timp. Totuși, în prezent, cloud computing se referă la numeroasele tipuri de servicii și aplicații livrate în cloud prin internet și la faptul că în multe cazuri dispozitivele utilizate pentru a accesa aceste servicii și aplicații nu necesită aplicații speciale. Articolul din studiu [4] introduce noțiunile de bază specifice cloud-ului, caracteristicile, modelele de servicii, modelele de implementare, beneficiile, provocările. De asemenea, se oferă o scurtă trecere în revistă a comunicațiilor din Cloud, punându-se accentul pe serviciile de comunicații, API-urile web, interfețele de control al serverului media și scalabilitatea comunicațiilor.

În acest context, [19] susține că Cloud Computing se referă atât la aplicațiile furnizate ca servicii pe Internet, cât și la sisteme software și hardware din centrele de date care furnizează aceste servicii. Serviciile în sine au fost mult timp denumite Software ca serviciu (SaaS). Hardware-ul și software-ul datacenterului sunt ceea ce vom numi Cloud. Atunci când un cloud este pus la dispoziția publicului larg, într-o manieră plătită în mod curent, îl numim Cloud Public; serviciul vândut este Utility Computing. Folosim termenul Private Cloud pentru a ne referi la centrele de date interne ale unei companii sau ale unei alte organizații, care nu sunt puse la dispoziția publicului larg. Astfel, Cloud Computing este suma SaaS și Utility Computing, dar nu include Cloud-urile private. Oamenii pot fi utilizatori sau furnizori de SaaS sau utilizatori sau furnizori de Utility Component. Ne concentrăm pe furnizorii SaaS (utilizatorii Cloud) și furnizorii de servicii Cloud, care au primit mai puțină atenție decât utilizatorii SaaS. Din punct de vedere hardware, trei aspecte sunt noi în Cloud Computing: Iluzia resurselor infinite de calcul disponibile la cerere, Eliminarea unui angajament up-front de către utilizatorii Cloud și Abilitatea de a plăti pentru utilizarea resurselor de calcul după cum este necesar.

[20], [21] oferă, de asemenea, un studiu destul de obiectiv privind tehnologiile și tendințele actuale legate de Cloud Computing și Data Storage și oferă, în același timp, o comparație profundă între avantaje și dezavantaje, precum și punctele de vedere al utilizatorului și ale furnizorului.

Zhang et al. concluzionează că cloud computing-ul a apărut în sfârșit ca o paradigmă convingătoare pentru gestionarea și furnizarea de servicii prin Internet în [3]. În același timp, în pofida beneficiilor semnificative oferite de cloud computing, tehnologiile actuale nu sunt suficient de mature pentru a-și realiza întregul potențial. Multe provocări cheie în acest domeniu, inclusiv furnizarea automată a resurselor, gestionarea energiei și gestionarea securității, încep să primească atenția comunității de cercetare. Lucrarea a analizat stadiul tehnicii cloud computing, acoperind concepțele sale esențiale, modelele arhitecturale, caracteristicile proeminente, tehnologiile cheie și direcțiile de cercetare. Dat fiind faptul că dezvoltarea tehnologiei cloud computing este încă într-o fază incipientă, se speră ca acest articol va oferi o mai bună înțelegere a provocărilor de proiectare din domeniul cloud computing și va deschide calea cercetărilor viitoare în acest domeniu.

3.2.2 Stocare și baze de date în Cloud

În prezent, este comună accesarea conținutului pe internet în mod independent, fără a se face referire la infrastructura de găzduire. Această infrastructură constă în centre de date care sunt monitorizate și întreținute în permanență de furnizorii de conținut. Cloud computing este o extensie a acestei paradigmă în care capabilitățile aplicațiilor de business sunt expuse ca servicii sofisticate care pot fi accesate printr-o rețea. Furnizorii de servicii de tip cloud sunt stimulați de profiturile care pot fi făcute prin tarifarea consumatorilor pentru accesarea acestor servicii. Consumatorii, cum ar fi întreprinderile, sunt atrași de oportunitatea de reducere sau eliminare a costurilor asociate furnizării acestor servicii "in-house". Cu toate acestea, deoarece aplicațiile cloud pot fi vitale pentru operațiunile de bază ale consumatorilor, este esențial ca aceștia să aibă garanții din partea furnizorilor cu privire la furnizarea serviciilor. În mod tipic, acestea sunt furnizate prin acorduri de nivel de servicii (SLA) intermediate între furnizori și consumatori.

Lucrarea [22] prezintă tendințele actuale din spațiul Cloud Computing și prezintă candidații pentru viitoarele îmbunătățiri ale acestei tehnologii: prezentarea viziunii secolului 21 în termeni de sisteme de calcul și descriere a diferitelor paradigmă de calcul care au promis sau promit să aducă această mare viziune; diferențiind Cloud computing de alte două paradigmă computerizate: Cluster computing și Grid computing; concentrându-se asupra serviciilor Cloud centrate pe VM și prezentând o arhitectură pentru crearea de nori orientați spre piață utilizând VM; oferind perspective asupra strategiilor de gestionare a resurselor bazate pe piață, care includ atât managementul serviciilor bazate pe client, cât și gestionarea riscurilor computaționale pentru a susține alocarea resurselor orientate spre SLA; dezvăluind gândurile inițiale cu privire la interconectarea cloud-urilor pentru crearea unui schimb global Cloud și compararea unor platforme reprezentative Cloud, în special a celor dezvoltate în industrii, împreună cu tehnologia Cloud de la compania Aneka. De asemenea, ne prezintă activitatea curentă privind Cloud Computing, care include realizarea alocării resurselor orientate spre piață a tehnologiilor cloud, aşa cum a fost realizat Aneka pentru întreprinderi business și subliniind diferența dintre volumul de lucru HPC și volumul de muncă pe Internet; încorporând o infrastructură de meta-negociere pentru managementul QoS pentru a stabili schimburile și piețele globale de cloud și a crea servicii cloud 3rd party bazate pe furnizarea de conținut de înaltă performanță pe serviciile de stocare cloud comerciale.

Cu cloud computing, scopul este de a ascunde complexitatea managementului infrastructurii IT de utilizatorii săi. În același timp, platformele de cloud computing oferă o scalabilitate masivă, o fiabilitate de 99,999 %, performanță ridicată și configurabilitate specificabilă. Aceste capacitațи sunt furnizate la costuri relativ scăzute în comparație cu infrastructurile dedicate. Articolele [23] și [24] oferă o introducere rapidă în spațiul de stocare în cloud. Aceasta acoperă tehnologiile cheie din Cloud Computing și Cloud Storage, mai multe tipuri de servicii de cloud și descrie avantajele și provocările Cloud Storage după introducerea modelului de referință Cloud Storage.

Cerințele de calcul și de stocare a aplicațiilor, cum ar fi cele pentru Big Data Ana-

3. Studiu bibliografic

lytics, IoT, Business Intelligence și rețele sociale, prin seturi de date peta-byte, au împins bazele de date centralizate, bazate pe SQL la limitele lor [25]. Acest lucru a dus la dezvoltarea unor baze de date orizontal scalabile, non-relaționale, distribuite, denumite NoSQL(Not only SQL). Între utilizări ale bazelor de date NoSQL, se amintesc: Procesarea de date pe scară largă (procesare paralelă pe sisteme distribuite); Embedded IR (informații de bază despre mașină-mașină; recuperare a datelor); Analiza exploratorie privind datele semi-structurate (nivel expert); stocare de date cu volum mare (nestructurat, semi-structurat, structurat cu pachete mici). NoSQL este un domeniu vast și extins, în scopul acestei lucrări - caracteristici (caracteristici și beneficii ale bazelor de date NoSQL); clasificare (4 categorii bazate pe caracteristicile acestora); comparație și evaluare (cu o matrice bazată pe câteva atribute - design, integritate, indexare, distribuție, sistem) a diferitelor tipuri de baze de date NoSQL; starea actuală de adoptare a bazelor de date NoSQL. Studiul [26] încearcă să ofere o înțelegere independentă a punctelor forte și a punctelor slabe ale diferitelor abordări baze de date NoSQL pentru a sprijini aplicațiile care procesează volume imense de date, precum și pentru a oferi o imagine de ansamblu globală a acestor baze de date non-relaționale NoSQL.

Progresul tehnologiei Web și proliferarea dispozitivelor mobile și senzorilor conectați la Internet au dus la cerințe imense de procesare și stocare. Cloud computing a apărut ca o paradigmă care promite să îndeplinească aceste cerințe. Lucrarea [26] se concentrează asupra aspectelor de stocare a cloud computing-ului, în special asupra gestionării datelor în medii cloud. Bazele de date relaționale tradiționale au fost concepute într-o eră hardware și software diferite și se confruntă cu provocări în îndeplinirea cerințelor de performanță și scalare ale Big Data. Bazele de date NoSQL și NewSQL se prezintă ca alternative care pot gestiona un volum imens de date. Datorită numărului mare și diversității soluțiilor NoSQL și NewSQL existente, este dificil să înțelegem domeniul și chiar mai dificil să alegem soluția potrivită pentru o anumită sarcină.

Articolul [27] analizează soluțiile NoSQL și NewSQL cu scopul de: (1) a oferi o perspectivă în domeniu, (2) a oferi îndrumări practicanților și cercetătorilor pentru a alege o bază de date adecvată (3) a identifica provocările și oportunitățile din domeniu. Mai exact, soluțiile cele mai proeminente sunt comparate, concentrându-se pe modele de date, interogări, scalabilitate și capabilități legate de securitate. Caracteristicile care conduc capacitatea de a scala cererile de citire și de a scrie sau de a scala datele de stocare sunt investigate, în special partitioarea, replicarea, consistența și controlul concurenței. Mai mult decât atât, sunt utilizate studiile de caz și scenariile în care au fost testate baze de date NoSQL și NewSQL și este examinată fezabilitatea diferitelor soluții pentru diferite seturi de aplicații. În același timp, s-au identificat provocările în acest domeniu, incluzând imensa diversitate și inconsistența terminologilor, documentația limitată, criteriile de comparare redusă și benchmark-uri și lipsa de limbaje de interogare standardizate. [28] prezintă modelul relational SQL în ceea ce privește matricele asociative și identifică proprietățile matematice cheie ale NewSQL și noSQL care sunt păstrate în cadrul SQL. Aceste proprietăți includ asociativitatea, comutativitatea, distributivitatea, identitățile, anihilatoarele și inversele. Măsurările de performanță privind distributivitatea și asoci-

ativitatea arată impactul pe care aceste proprietăți îl poate avea asupra operațiilor asociative de matrice. Aceste rezultate demonstrează că matricea asociativă poate furniza un model pentru ca polystores să utilizeze proprietățile matematice în cadrul bazelor de date pentru a optimiza schimbul de date și interogări.

Furnizori precum Amazon, Google, Salesforce, IBM, Microsoft și Sun Microsystems au început să creeze noi centre de date pentru găzduirea aplicațiilor Cloud computing în diverse locații din întreaga lume pentru a oferi redundanță și a asigura fiabilitatea în caz de defecțiuni la nivel de site. Întrucât cerințele utilizatorilor pentru serviciile cloud sunt variate, furnizorii de servicii trebuie să se asigure că pot fi flexibili în furnizarea serviciilor, păstrând în același timp utilizatorii izolați de infrastructura de bază. Progresele recente în tehnologia și software-ul microprocesoarelor au condus la creșterea capacitații hardware-ului de mărfuri de a rula eficient aplicațiile în mașinile virtuale (VM). VM permite atât izolarea aplicațiilor de la hardware-ul de bază, cât și celealte VM-uri, precum și personalizarea platformei pentru a se potrivi nevoilor utilizatorului final. Furnizorii pot expune aplicații care rulează în VM-uri sau pot oferi acces la VM-uri ca un serviciu (de exemplu, Amazon Elastic Compute Cloud), permitând astfel consumatorilor să instaleze propriile aplicații. În timp ce este convenabil, utilizarea VM-urilor provoacă provocări suplimentare, cum ar fi alocarea inteligentă a resurselor fizice pentru gestionarea cererilor de resurse concurente ale utilizatorilor. În plus, consumatorii de servicii pentru întreprinderi cu operațiuni globale necesită un timp de răspuns mai rapid și, prin urmare, economisesc timp prin distribuirea solicitărilor de încărcare infrastructurii cloud multiple în diferite locații în același timp. Acest lucru determină nevoie de a crea o atmosferă de calcul pentru interconectarea dinamică și furnizarea de soluții cloud din mai multe domenii în cadrul și între întreprinderi. Există numeroase provocări legate de crearea unor astfel de interconexiuni la cloud la cloud.

3.3 Realitate augmentată

3.3.1 Generalități, definiții și concepte în AR

Realitatea augmentată (AR - Augmented Reality) este o variație a mediilor virtuale (VE - Virtual Environments) sau Realitatea virtuală, așa cum este definită mai des. Astăzi, aceste 2 domenii sunt deseori tratate separat. În realitatea virtuală, utilizatorul este cufundat într-un mediu sintetic. În timp, în acest mediu, utilizatorul nu este conștient de lumea din jurul lui. În schimb, AR permite utilizatorului să percepă lumea reală cu obiecte virtuale suprapuse sau compuse cu lumea reală. Prin urmare, AR suplimentează realitatea, în loc să o înlocuiască complet. Unii cercetători definesc AR într-un mod care necesită utilizarea HMD (Head-Mounted-Display). Azuma et al. în [5] studiază stadiul actual al realității augmentate și descrie lucrarea efectuată în multe site-uri diferite și explică problemele apărute la momentul construirii sistemelor Augmented Reality și de asemenea rezumă compromisurile și abordările luate până în prezent pentru a depăși aceste probleme

3. Studiu bibliografic

și speculează direcțiile viitoare care merită explorate. Se prezintă, de asemenea, câteva aspecte care apar atunci când se dezvoltă un sistem de realitate augmentată, mai precis, două dintre ele fiind înregistrarea și simțirea.

Clasele de potențiale aplicații AR explorate sunt următoarele: vizualizarea medicală, întreținerea și repararea, adnotarea, planificarea traseelor robotului, divertismentul și navigarea și întâmpinarea aeronavelor militare.

Doctorii ar putea utiliza AR ca ajutor de vizualizare și de formare sau asistent pentru intervenții chirurgicale. Instrucțiunile vizuale reamintesc unui chirurg nou pașii necesari, fără a fi nevoie să consulte un manual. Virtual, se pot identifica, de asemenea, organe și specifica locații pentru a evita perturbarea organelor sensibile, așa cum este menționat în documentul [29]. Grupurile de la MIT AI Lab [30], General Electric [31], și în altă parte [32] se concentreză pe afișarea datelor RMN sau CT, înregistrate direct pe pacient.

O altă categorie de aplicații Augmented Reality este asamblarea, întreținerea și repararea mașinilor complexe. Instrucțiunile sunt mai ușor de înțeles dacă sunt disponibile ca obiecte 3D sau desene tehnice suprapuse peste echipamentul propriu-zis, prezentând pas cu pas sarcinile care trebuie făcute și cum trebuie făcute. Animăriile sunt, de asemenea, utile în acest context, făcând indicațiile și instrucțiunile mai explicite. Unele prototipuri există deja în acest domeniu, [33].

Tehnicile de adnotare și vizualizare folosite de AR pot fi folosite pentru obiecte și medii cu informații publice sau private. De exemplu, un afișaj portabil ar putea furniza informații despre conținutul rafturilor de bibliotecă pe măsură ce utilizatorul umblă în jurul bibliotecii [34].

Teleoperarea unui robot este adesea o problemă dificilă, mai ales atunci când robotul este departe, cu întârzieri lungi în legătura de comunicare. În această situație, în loc de a controla direct robotul, ar fi preferabil să se controleze o versiune virtuală a robotului. Sistemul ARGOS a demonstrat că stereoscopic AR este un mod mai ușor și mai precis de a face planificarea traseului robot decât interfețele monoscopice tradiționale [35]. Alții au folosit, de asemenea, suprapuneri înregistrate cu sisteme de teleprezentare [36].

În divertisment, împreună cu explozia tehnologică a dispozitivelor portabile și a dispozitivelor mobile, există o multitudine de aplicații, jocuri și soluții de afaceri care oferă deja aplicații specifice realității augmentate.

Avioanele militare și elicopterele au utilizat afișaje pentru cap și hover (HUD) și vizoare montate pe căști (HMD) pentru a suprapune grafica vectorială în viziunea pilotului asupra lumii reale.

Ingineria civilă, arhitectura și amenajările interioare au produs propriile aplicații pentru uz comercial. Clientii pot avea o senzație mai bună despre modul în care proiectarea casei lor sau modul în care casele lor pot arăta pe anumite terenuri. Realitatea augmentată este mult în spatele mediilor virtuale în maturitate. Mai mulți furnizori comerciali vând sisteme integrate complete pentru mediul virtual. Cu toate acestea, foarte puțini furnizori vând în prezent un sistem HMD bazat pe Realitatea Augmentată. Câteva seturi virtuale de sistem bazate pe monitor sunt disponibile, dar astăzi majoritatea sistemelor AR sunt

întâlnite în laboratoarele de cercetare academică și industrială.

3.3.2 Directii de studiu și tehnologii

Numeroase lucrări de cercetare s-au îndreptat spre realitatea augmentată, iar educația și mediile academice sunt direcții importante de experimentare în acest sens. Lucrearea [37] acoperă o imagine de ansamblu a AR prin analiza definițiilor, taxonomiilor și tehnologiilor AR. Punctul principal gravitează în jurul vizualizării AR ca concept, mai degrabă decât a unui tip de tehnologie, prin identificarea anumitor caracteristici și "affordances" ale sistemelor și aplicațiilor AR. Apoi, se face o analogie, afirmând că aceste caracteristici nu pot fi unice pentru aplicațiile AR și că pot fi găsite în alte sisteme tehnologice sau medii de învățare (de exemplu, medii de învățare omniprezente și mobile). Abordarea de instruire adoptată de un sistem AR și alinierea dintre proiectarea tehnologică, abordarea instruirii și experiențele de învățare pot fi mai importante. Astfel, se clasifică trei categorii de abordări instructive care subliniază "rolurile", "sarcinile" și "locațiile" și se discută despre ce și cum pot aborda diferențele categorii de AR, pentru a ajuta elevii să învețe. Anumite aspecte tehnologice și pedagogice, de învățare, sunt de asemenea explorate. [38], [39] și [40] se concentrează pe conceptele de "affordances" ale realității augmentate în învățare, dar, bineînțeles, au diferențe abordări legate fie de fizică [41], matematică, biologie [42], sau alte științe exacte[43]. Altele, precum [43] și [44], se concentrează pe caracteristicile intangibile ale studenților, adică motivarea studenților prin oferirea de cursuri vizuale. În cele din urmă, unele directii de studiu, precum [45] se bazează pe integrarea realității augmentate în modelele arhitecturale vizualizate.

Pe de altă parte, există cercetări care se concentrează mai mult pe aspectele practice ale realității augmentate. Rezultatul final al [46] este un sistem de ghidare mobil care integrează instrucțiuni de apreciere a artei cu realitatea augmentată (AR) și a fost conceput ca un instrument auxiliar pentru aprecierea picturii și performanței de învățare a trei grupuri de participanți vizitatori: ghidați prin AR, ghidați prin audio și fără ghid. Rezultatele au arătat că, în comparație cu participanții audio și non-ghidați, ghidul AR a îmbunătățit în mod considerabil eficacitatea învățării vizitatorilor, a promovat experiența fluxului și a extins timpul petrecut de vizitatori concentrându-se asupra picturilor.

Mai mult, [47] descrie un sistem experimental de realitate augmentată mobil (MARS) pentru interfețe-utilizator ce permite utilizatorilor din exterior și interior să acceseze și să gestioneze informațiile înregistrate spațial în lumea reală. Utilizatorii externi pot experimenta prezentări multimedia în spațiu care sunt reprezentate pe un afișaj HMD (head-mounted-display) purtat pe cap, utilizat împreună cu un computer-stilou. Utilizatorii interni pot obține o imagine de ansamblu a scenei în aer liber și pot comunica cu utilizatorii externi printr-o interfață de utilizator pentru desktop sau printr-o interfață utilizator "head-" și "hand-tracked".

Zonele urbane și AR sunt, de asemenea, obiectul studiului în [48], și [49] care au dus crearea unu prototip mașină touring.

[50] oferă un studiu comparativ al diferențelor kituri software de AR (SDK) dispo-

3. Studiu bibliografic

nibile pentru a crea aplicații de realitate augmentată. Lucrarea descrie modul în care realitatea augmentată este diferită de realitatea virtuală; funcționarea sistemului de realitate augmentată și diferite tipuri de urmărire folosite în AR în timp ce [51] și [52] conțin implementări concrete.

Capitolul 4

Fundamente Teoretice

4.1 Internet-of-Things

Internetul "lucrurilor" sau "Internet-of-Things" (IoT) se referă la dispozitivele conectate în mod intelligent și sisteme capabile de a gestiona date achiziționate prin senzori embedded și actuatori care aparțin anumitor mașinării sau obiecte. Se așteaptă ca IoT să aibă o expansiune extrem de rapidă în anii ce urmează iar aceasta convergență va oferi oportunități imense pentru servicii care vor îmbunătății radical calitatea vieții pentru consumatori și producători deopotrivă.

Ideea inițială a IoT a fost propusă de către Laboratoarele Auto-ID de la MIT (Massachusetts Institute of Technology) la sfârșitul anilor 1990([9]) mai mult din nevoi de ordin logistic. Rapoartele ITU 2005([9]) indică faptul că ne îndreptăm spre o societate omniprezentă ("ubiquitous network society"). Prin aceasta se face referire la faptul că rețelele și dispozitivele interconectate sunt omniprezente în viața de zi cu zi. IoT este în strânsă legătură cu Internetul, rețelele mobile de comunicare și rețelele de senzori wireless.

Cu toate acestea însă, nu există o definiție singulară, universal valabilă și acceptată pentru sintagma "Internet of Things". Diferite grupuri descriu în mod diferit definițiile pentru a promova un anumit punct de vedere sau opinie a ceea ce IoT înseamnă și care sunt cele mai importante atrbute.

De exemplu, în studiul [53], The Internet Architecture Board (IAB) oferă urmatoarea descriere:

The term "Internet of Things" (IoT) denotes a trend where a large number of embedded devices employ communication services offered by the Internet protocols. Many of these devices, often called "smart objects", are not directly operated by humans, but exist as components in buildings or vehicles, or are spread out in the environment.

In cercul Internet Engineering Task Force(IETF), termenul "rețele de obiecte inteligente" este deseori folosit în referință cu Internet of Things. În acest context, obiectele inteligente sau "smart objects" sunt dispozitive care au conținut semnificativ, cum ar

4. Fundamente Teoretice

fi putere limitată, memorie, resurse de procesare și lungimea de bandă. Munca în IETF este organizată în specificații diferite în vederea realizării interoperației între diferite tipuri de obiecte inteligente.

Definiția într-o din edițiile IEEE Communications Magazine face asocierea logică între IoT cu servicii cloud:

The Internet of Things (IoT) is a framework in which all things have a representation and a presence in the Internet. More specifically, the Internet of Things aims at offering new applications and services bridging the physical and virtual worlds, in which Machine-to-Machine (M2M) communications represents the baseline communication that enables the interactions between Things and applications in the cloud.

Toate definițiile descriu scenarii în care conectivitatea rețelelor și puterea computațională se extinde spre a încorpora constelații de obiecte, dispozitive, senzori și itemi care în mod normal nu ar fi considerate "calculatoare"; aceasta permite dispozitivelor să genereze, comunice și să consume date, de multe ori cu intervenție umană minimă.

4.1.1 Modele de comunicare în Internet of Things

Comunicare Dispozitiv-spre-Dispozitiv

Modelul de comunicare device-to-device reprezintă două sau mai multe dispozitive care conectează și comunică între ele și nu printr-un application server intermediar. Aceste dispozitive comunică prin mai multe tipuri de rețele, inclusiv rețele IP sau internet. Cu toate acestea, deseori, aceste dispozitive folosesc protocoale Bluetooth, Z-Wave sau ZigBee pentru a stabili o comunicare directă dispozitiv-spre-dispozitiv, așa cum este prezentată în 4.1.



Figura 4.1: Modelul device-to-device conform [1]

Aceste rețele permit dispozitivelor să adere la un protocol de comunicare specific pentru a comunica și să schimbe mesaje pentru a-și atinge scopul. Modelul de comunicare este folosit în aplicații cum ar fi sistemele "home-automation", care folosesc pachete mici de informație.

Comunicare Dispozitiv-spre-Cloud

Într-un model de comunicare device-spre-Cloud, dispozitivul IoT se conectează la un serviciu cloud pentru a interschimba date și a controla traficul de mesaje. Această abordare beneficiază de mecanismele de comunicare deja existente cum ar fi conexiuni Ethernet sau Wi-Fi pentru a realiza o conexiune între dispozitive și rețele IP, care în final se conectează la un serviciu cloud. Pe lângă aceasta, această conexiune oferă posibilitatea utilizatorului să acceseze termostatul de la distanță.

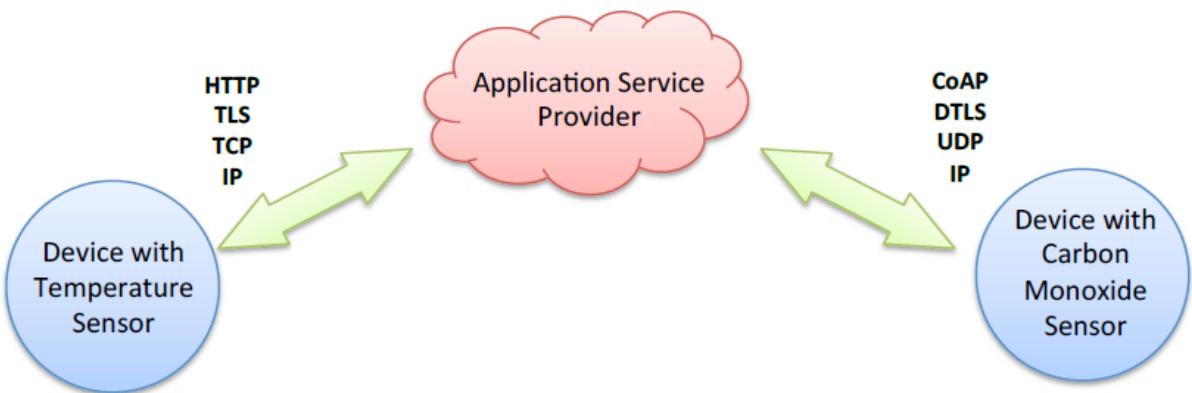


Figura 4.2: Modelul device-to-cloud conform [1]

Modelul de comunicare adoptat de cele mai populare dispozitive IoT, cum ar fi Nest Labs Learning Thermostat și Samsung SmartTV. În cazul Nest Learning Thermostat, dispozitivul transmite datele într-o bază de date în cloud unde poate fi analizat consumul de curent. În mod similar, televizorul Samsung transmite printr-o conexiune la internet informații cu privire la preferințele utilizatorului și în același măsură interacționează cu privitorul prin serviciul de recunoaștere vocală. Se poate observa totuși că în marea majoritate a cazurilor, dispozitivul și serviciul cloud provin de la același producător, aceasta ducând la anumite provocări referitoare la interoperabilitate. Acest model este reprezentat în 4.2.

Comunicare Device-to-Gateway

În modelul dispozitiv-spre-gateway, sau "device-to-application-layer gateway" (ALG) model, dispozitivul IoT se conectează prin un serviciu ALG care funcționează ca un relee pentru a ajunge la un serviciu cloud. În termeni simpli, aceasta înseamnă că există aplicație software pe un dispozitiv gateway care are rolul de intermediar între dispozitiv și cloud oferind pe lângă multe altele, securitate și translație a datelor și a protocolelor. Modelul poate fi văzut în Figura 4.3.

Câteva variații ale acestui model pot fi regăsite în dispozitive de consum. În multe cazuri, dispozitivul local gateway este un smartphone care rulează o aplicație pentru a comunica și direcționa date spre un serviciu cloud. Acesta este cel mai des întâlnit model

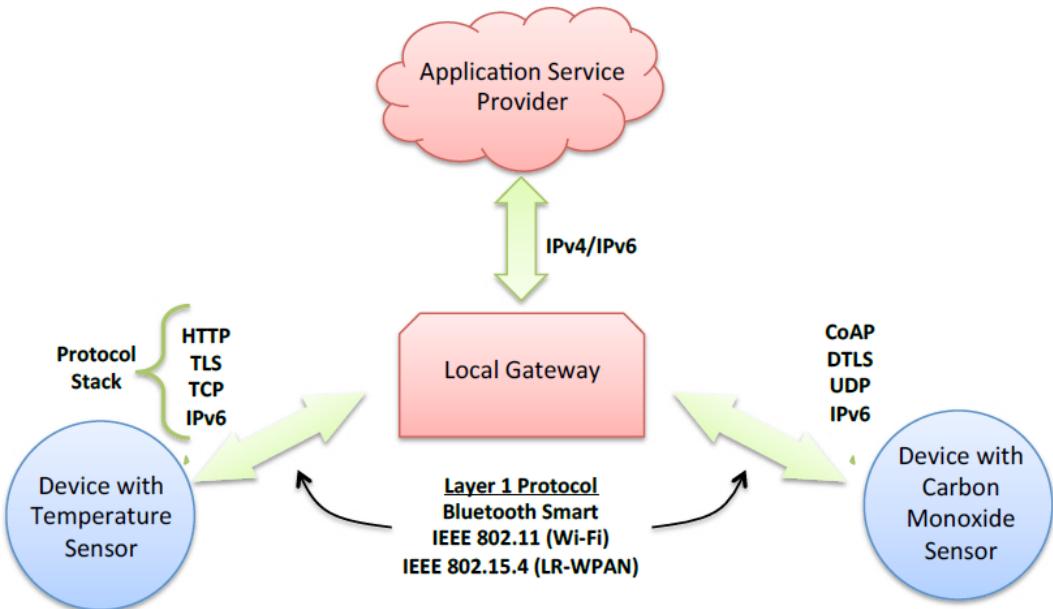


Figura 4.3: Modelul device-to-gate conform [1]

printre produsele de cosum cum ar fi trackere/brățările fitness. Aceste dispozitive nu au abilitatea nativă de a se conecta direct la un serviciu cloud, și de multe ori, direcționează informațiile spre o aplicație care rulează pe un smartphone-uri, aceasta constituind un gateway care conectează dispozitivul la cloud.

În altă formă, dar aparținând aceluiași model, este trend-ul "hub-urilor" în aplicațiile de home automation. Pe lângă faptul că reprezintă un gateway, de asemenea pot umple anumite goluri de interoperabilitate între dispozitive.

Acest model de comunicare este în mod frecvent folosit pentru a integra noi dispozitive "smart" într-un sistem legacy cu dispozitive care nu operează nativ între ele. Un dezavantaj al acestei abordări este faptul că se adaugă un nou nivel de complexitate și cost în tot sistemul.

Comunicare Back-End Data-Sharing

Acest model se referă la o arhitectură de comunicare care determină utilizatorii să exporte și să analizeze datele de la obiectele inteligente dintr-un serviciu cloud în combinație cu date de la alte surse. Această arhitectură permite ca stream-urile de date colectate de la un singur dispozitiv să fie aggregate și analizate. De asemenea, acest model sugerează un model cloud, numit "federated cloud services" sau cloud API care realizează interoperabilitatea cu datele stocate în cloud. Figura 4.4 reprezintă acest model.

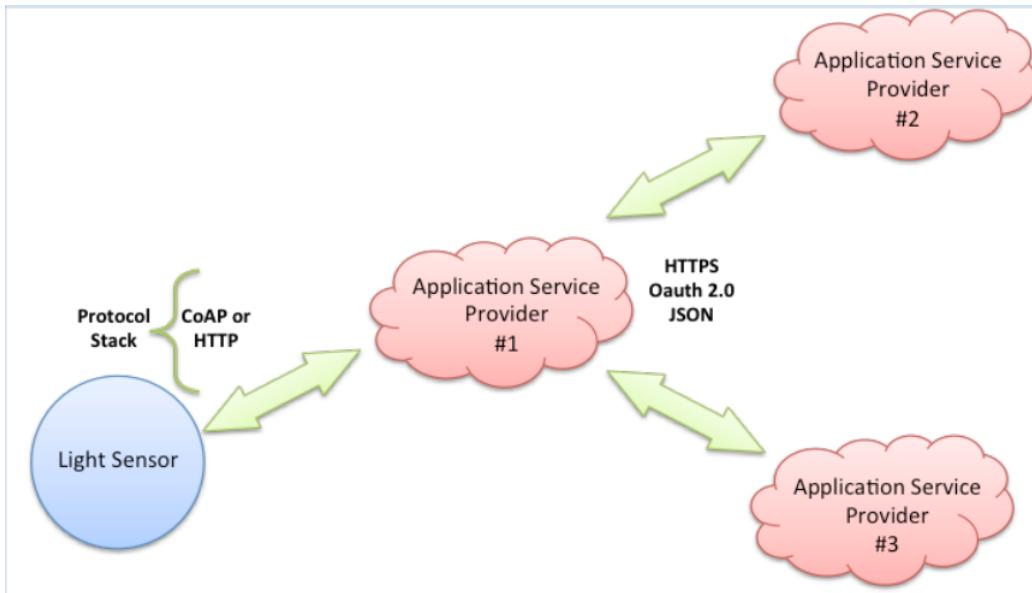


Figura 4.4: Modelul back-end data sharing model conform [1]

IoT ca o Rețea de Rețele

În forma actuală, IoT este construit dintr-o colecție de rețele disjuncte și specifice. Mașinile moderne conțin rețele multiple care controlează funcționalitatea motorului, sistemele de securitate, sistemele de comunicare și aşa mai departe. Clădirile comerciale și rezidențiale de asemenea au sisteme de control climaterice, de ventilare și aer condiționat (HVAC), servicii telefonice, securitate și iluminare. Pe măsură ce IoT evoluează, aceste rețele, precum și multe altele vor fi interconectate, ca în 4.5.

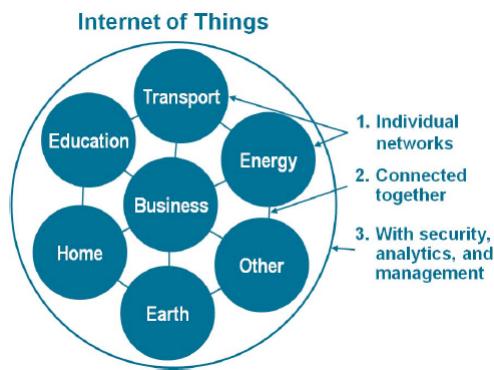


Figura 4.5: IoT ca rețea de rețele conform [2]

4.2 Infrastructura Cloud

Termenul de "cloud" are originile în diagramele rețelelor care reprezintă internetul sau anumite componente care fac parte din el, cum ar fi "schematic clouds". Termenul de "cloud-computing" a fost inventat pentru a sublinia ceea ce se întâmplă când aplicațiile și serviciile sunt mutate în "cloud". "Cloud Computing" nu reprezintă un concept inventat peste noapte. Unele trăsături pot fi urmărite în timp la momentul în care sistemele de calcul și unitățile de procesare aveau acces comun într-un timp limitat la resursele și aplicații. În prezent, totuși, acest termen de "Cloud Computing" se referă la diferitele tipuri de servicii și aplicații livrate în "internet cloud" și la faptul că, în majoritatea cazurilor, dispozitivele care accesează aceste servicii nu necesită aplicații speciale.

Caracteristici principale:

- Infrastructură comună - folosind un model software virtualizat, facilitând accesul comun la serviciile fizice și capabilitățile rețelistiche. Infrastructura cloud, indiferent de modelul de instalare, are ca scop valorificare optimă a infrastructurii disponibile pentru un număr limitat de utilizatori;
- Gestiunea dinamică a resurselor (Dynamic Provisioning) - permite gestiunea serviciilor în funcție de cerințele și specificațiile curente. Aceasta se face implicit prin software automatizat, permitând extinderea sau constrângerea capabilităților unui serviciu, după nevoi. Această scalare dinamică este necesară pentru a menține nivele înalte de securitate și fiabilitate;
- Accesibil prin rețele - serviciile se accesează prin internet de pe diferite dispozitive folosind API-uri standardizate. Configurarea și instalarea serviciilor în cloud poate include aplicații dintr-o gamă extrem variată, pornind de la aplicații business până la ultimele app-uri de pe smartphone-uri;
- Gestionarea măsuratorilor (Managed Metering) - utilizarea metricilor pentru gestiunea și optimizarea serviciului și pentru a obține informații referitoare la costul serviciului. În acest fel, consumatorii sunt facturați pentru servicii în funcție de cât folosesc în această perioadă;

În mod general, arhitectura unei infrastructuri sau mediu cloud poate fi împărțită în 4 nivele logice: nivelul Hardware/Datacenter, nivelul de infrastructură, nivelul de platformă și nivelul de aplicație așa cum este prezentat în Figura 4.6 .

Nivelul de hardware

Acest nivel este responsabil cu gestiunea resurselor fizice ale cloud-ului, cum ar fi serverele fizice, router-ele, switch-urile, sistemele de alimentare și răcire. În practică, acest nivel este implementat în centre de date. Un centru de date (data center) conține în mod

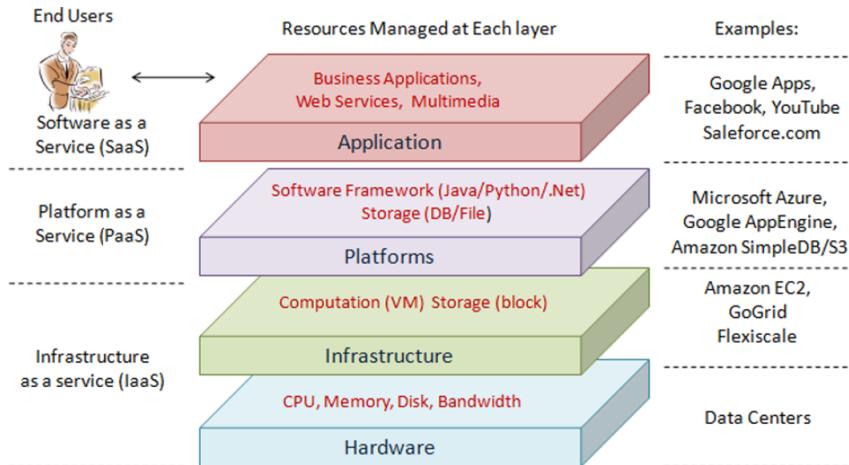


Figura 4.6: Arhitectura unui cloud conform [3]

normal mii de servere care sunt organizate în rack-uri și interconectate prin switch-uri, routere și alte componente. Problemele frecvente la acest nivel țin în mod normal de configurația hardware-ului, toleranță la defecte, gestiunea traficului, puterii de consum și răcire.

Nivelul de infrastructură

De asemenea este cunoscut ca și nivelul de virtualizare, acest nivel crează un ”pool” de resurse de stocare și calcul, partitioanând resursele fizice folosind tehnologii de virtualizare (KVM, VMware, etc.). Nivelul de infrastructură este o componentă esențială, devreme ce caracteristici-cheie, cum ar fi alocarea dinamică a resurselor este disponibilă prin tehnologii de virtualizare.

Nivelul de platformă

Construit peste nivelul de infrastructură, acest nivel conține sistemele de operare și framework-urile aplicațiilor. Scopul acestui nivel este de a minimiza efortul de a configura și rula aplicațiile direct în container-ele VM. De exemplu, Google App Engine operează la nivelul de platformă pentru a oferi suport API pentru implementarea aplicațiilor de stocare, baze de date și logica de business a unei aplicații web tipice.

Nivelul de aplicație

La cel mai înalt nivel ierarhic, nivelul de aplicație constituie suportul pentru aplicațiile în sine. O diferențiere majoră față de abordările clasice o constituie faptul că aplicațiile cloud beneficiază de scalabilitatea automată pentru obținerea performanței, disponibilitatea și costul redus de operare.

4. Fundamente Teoretice

Modelele principale de servicii de servicii implementate sunt următoarele:

- Software as a Service (SaaS) - Consumatorii achiziționează posibilitatea de a accesa și utiliza o aplicație sau un serviciu care este host-at în Cloud. Un exemplu pentru acest model este Salesforce.com, unde informația necesară pentru interacțiunea dintre consumator și serviciu este integrată ca și parte din acel serviciu. De asemenea, Microsoft a făcut pași importanți în această direcție, prin serviciu Office Web Apps disponibile sub diferite licențe și subscriskii.
- Platform as a Service(PaaS) - Consumatorii achiziționează accesul la aceste platforme, fiind capabili apoi de a-și lansa propriile lor soluții software și aplicații în cloud. Sistemele de operare și accesul la rețele nu este gestionat de către consumator întrucât pot exista constrângeri asupra tipurilor de aplicații permise sau suportate.
- Infrastructure as a Service (IaaS) - Consumatorii pot controla și gestiona sistemele cu privire la sistemele de operare, aplicații, stocare și conectivitatea la rețea dar fară a controla infrastructura cloud-ului.

În această privință, există și numeroase modele derivate din cele principale enumerate mai sus care sunt legate de un anumit tip de industrie. Un astfel de model este CaaS (Communication as a Service). Acest model se atribuie în principal serviciilor de telefonie IP. În acest context, CaaS poate fi văzut ca un subset al SaaS.

Modele de configurare și implementare

Cele 4 modele de lansare în cloud au fost identificate:

- Cloud Privat - infrastructura cloud a fost lansată și este gestionată și operată de către o organizație anume. Operarea poate fi "in-house" sau cu un terț "on-premise".
- Cloud Public - acest model este disponibil publicului pentru a fi comercializat de către un furnizor de soluții cloud. Acest lucru determină dezvoltatorul să dezvolte și să lanseze un serviciu în cloud fără investiții majore.
- Cloud Community - această infrastructură este comună pentru un număr de organizații cu interese și specificații similare. Costul poate fi redus în acest mod, devreme ce se poate distribui între membrii acestei comunități. Operarea poate fi "in-house" sau cu un terț "on-premise".
- Cloud hibrid - un număr de cloud-uri de orice tip dar aceste cloud-uri trebuie să permită migrarea aplicațiilor sau interfețelor de pe un cloud pe altul.

Figura 4.7 definește exemplele enumerate mai sus.

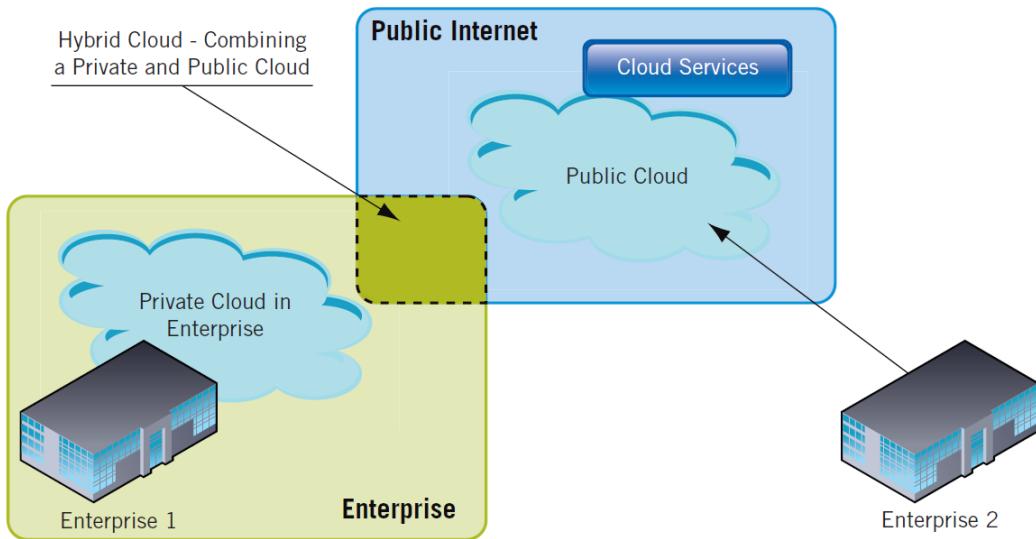


Figura 4.7: Exemplu de modele Cloud: Public, Privat, Hibrid preluat din [4]

4.2.1 Paradigma NoSQL

Serviciile de stocare în cloud reprezintă o componentă esențială și definitorie a paradigmelor de calcul în cloud, încrucișând nevoie ca informația folosită de aplicații să fie menținută cumva și undevară. Analize Big Data, data warehouse, Internet of Things, baze de date, backup și archive, toate persistă pe o arhitectură de stocare a datelor. Unele avantaje ale stocării în cloud o constituie fiabilitatea, scalabilitatea și securitatea datelor, în antiteză cu sistemele tradiționale de stocare "on-premise".

NoSQL(Not only SQL) cuprinde o varietate de tehnologii de baze de date care au fost concepute și dezvoltate ținând cont de cerințele și specificațiile aplicațiilor moderne:

- Dezvoltatorii lucrează cu aplicații care necesită crearea unor volume mari de date noi, cu tipuri de date variabile, structurate și semiestructurate, nestructurate, polimorfice.
- Echipe mici lucrează în metodologii de tip AGILE, în sprint-uri scurte, iterând rapid și urcând cod extrem de frecvent pe sistemele productive
- Aplicațiile care deserveau unui număr redus de consumatori, acum sunt livrate sub formă de servicii, accesibile de pe diferite dispozitive și scalate global spre milioane de utilizatori.
- Organizațiile se orientează spre arhitecturi care folosesc software open-source, commodity servers și cloud computing, spre deosebire de serverele laborioase infrastructuri de stocare locale și încete.

- Bazele de date relaționale nu au fost proiectate pentru a gestiona provocările referitoare la agilitate și scalabilitate cu care se confruntă marea majoritate a aplicațiilor moderne de azi.

NoSQL este un termen folosit pentru a descrie bazele de date ultra-performante, non-relaționale. Acestea utilizează o varietate de modele de date, dar cele mai importante sunt: modelul document, modelul graf, cheie-valoare și modelul columnar. Sunt recunoscute ca fiind scalabile, performante, cu o disponibilitate și fezabilitate ridicată. În lucrarea de față se va prezenta doar tipul document.

Tipul Document

Baza de date de tip-document a fost concepută pentru a stoca date semi-structurate, cum ar fi documente, de cele mai multe ori în format JSON (JavaScript Object Notation) sau XML (Extensible Markup Language). Spre deosebire de bazele de date tradiționale, cele reaționale, schema pentru fiecare document non-relațional (NoSQL) poate varia, oferind dezvoltatorilor, administratorilor de baze de date și profesioniștilor IT mai multă flexibilitate în organizarea și stocarea datelor aplicației și reducând spațiul de stocare necesar pentru valori optionale. Documentul JSON ușurează munca dezvoltatorilor pentru a serializa și încărca obiecte care conțin proprietăți și date relevante. Bazele de date NoSQL sunt proiectate pentru scalabilitate, folosindu-se de clustere distribuite și hardware low-cost care reduc latența și "throughput".

4.3 Realitatea Augmentată

Realitatea augmentată sau "Realitatea Virtuală Îmbunătățită" (AR) combină lumea reală cu cea digitală. În prezent, majoritatea cercetărilor utilizează imagini video live care sunt procesate digital pentru a adăuga grafică generată pe calculator. În alte cuvinte, sistemul augmentă imaginea cu date digitale.

Totodată, cercetarea cu privire la realitatea augmentată combină domeniile de Viziune computațională și Grafica pe calculator. În contextul AR, cercetarea din Computer Vision include, printre altele, detecția și urmărirea trăsăturilor bazate pe marker, detecția mișcării și urmărirea ei, analiza imaginilor, recunoașterea gesturilor și construcția unor mediilor controlate cu diferite tipuri de senzori. Grafica pe calculator include randarea obiectelor foto-realiste și animațiile interactive.

Cercetătorii definesc adesea realitatea augmentată ca un sistem în timp real. Cu toate acestea însă, se consideră augmentată și imaginile nemîșcate atât timp cât sistemul realizează fenomenul de "augmentare" în 3D și există o oarecare interacțiune la mijloc.

Tom Caudell, un cercetător la producătorul de aeronave Boeing, a definit termenul "augmented reality" în 1992. El a aplicat acest termen în mod special unui dispozitiv HMD (head-mounted-display) care ghida muncitorii în asamblarea firelor electrice la aeronave. Figura 4.8 prezintă sistemul HMD de realitate augmentată.

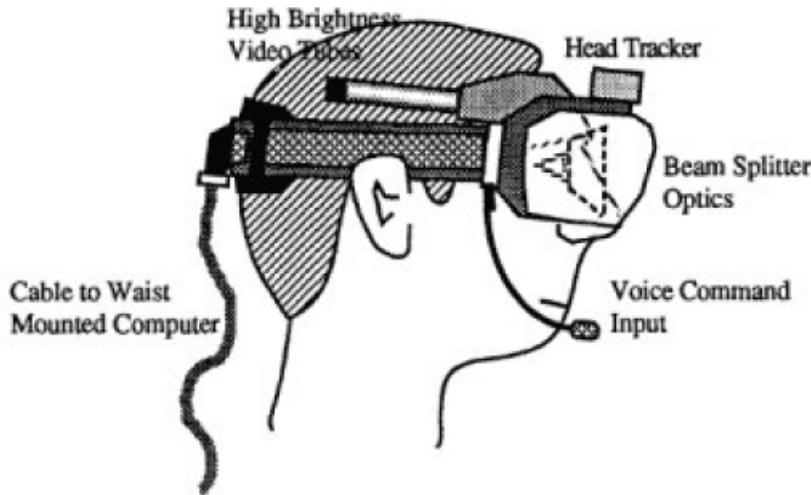


Figura 4.8: Model de head-mounted-display conform [5]

În anul 1994, Paul Milgram a prezentat *continuumul reality-virtuality*, care mai poartă numele de *continuumul mixed-reality*. Acest model constituie de fapt un segment care conține la un capăt mediul real iar la celălalt elemente de realitate virtuală, în timp ce tot ce e în mijloc constituie această realitate mixtă (Figura 4.9). Un sistem în realitatea mixtă combină lumea reală cu lumea virtuală pentru a produce un mediu nou unde obiectele fizice și cele digitale co-există și interacționează. Realitatea, în acest context, este reprezentată de mediul fizic care poate fi vizualizat, de exemplu, printr-un display video.



Figura 4.9: Continuumul Reality-Virtuality creat de Milgram din [6]

În anul 1997, Ronald Azuma a publicat un studiu extrem de cuprinsător asupra realității augmentate [5] și după o dezvoltare rapidă a domeniului, a lansat un nou studiu în 2001 [54]. El definește realitatea augmentată ca un sistem identificabil print 3 caracteristici:

1. combină realul cu virtualul;
2. este interactiv în timp real;
3. este înregistrat în 3D;

Aceasta este definiția pe care studiul prezent se bazează.

4.3.1 Realitatea augmentată simplă

Un sistem trivial de realitate augmentată este reprezentat de o cameră (cameră web), o unitate de calcul și un display(laptop). Camera captează imagine și sistemul aumentează obiectele virtuale deasupra unei imagini și afișează rezultatul. Figura 4.10 ilustrează un exemplu de sistem AR bazat pe marker. Sistemul capturează o imagine din mediu, detectează marker-ul, deduce și estimează orientarea și locația camerei iar apoi aumentează un obiect virtual deasupra imaginii și îl afișează pe ecran.



Figura 4.10: Exemplu de configurare a unui sistem AR din [6]

Figura 4.11 reprezintă un pipeline al unui sistem AR simplu. Modulul de captură preia imaginea de la cameră. Modulul de urmărire calculează locația și orientarea corectă pentru suprapunerea virtuală. Modulul de randare combină imaginea originală cu componentele virtuale folosind ”pose”-ul calculat și randează imaginea augmentată pe ecran.

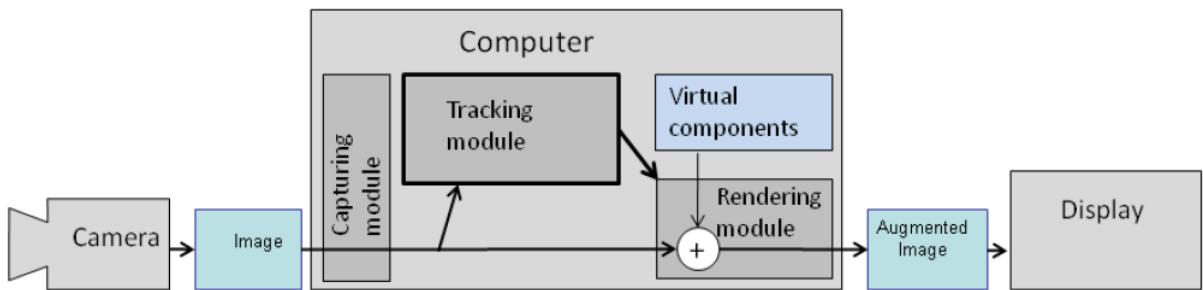


Figura 4.11: Organigramă unu sistem simplu AR, preluat din [6]

Modulul de urmărire reprezintă ”inima” sistemului de realitate augmentată; se calculează componenta *pose* relativă a camerei, în timp real. Termenul *pose* se referă la poziția celor ”six degrees of freedom” (DOF), i.e. locația 3D și orientarea 3D a unui obiect.

Modulul de urmărire determină sistemul să adauge componente virtuale ca și parte din scena reală. Diferența fundamentală în comparație cu alte unelte de procesare a imaginilor este că în AR obiectele virtuale sunt mutate și rotite într-un sistem de coordonate 3D, spre deosebire de coordonatele 2D ale imaginii.

Modul cel mai simplu de a calcula *pose* este de a utiliza markere. Cu toate acestea, modelul matematic din spatele metodelor de calcul a *pose*-ului este la fel ca și în alte situații din Computer Vision.

Achiziția imaginilor nu prezintă un interes major în Realitatea augmentată. De obicei, se utilizează o librărie de captură video deja integrată în librăria AR.

Modulul de randare desenează imaginea virtuală peste imaginea preluată de cameră. În grafica pe calculator, scena virtuală este proiectată pe un plan folosind o cameră virtuală iar apoi proiecția e randată. Calibrarea acestei camere este necesară ca parte a sistemului sau poate fi un proces separat. Multe unelte folosesc această componentă de calibrare, de ex. ALVAR și ARToolkit.

Varietatea de dispozitive posibile pentru un sistem de realitate augmentată este imensă. Începând de la PC, laptop, mini-PC, tablet PC, smartphone-uri sau alte unități de calcul. În funcție de aplicație, se pot utiliza diferite tipuri de camere. Pot fi amplasate prin HMD, see-through display, display extern sau display construit în unitatea de calcul. De asemenea, sistemul poate projecța partea augmentată sau poate folosi un display stereo.

4.3.2 Detectia bazata pe marker

Deoarece informația se prezintă într-un context al lumii reale, sistemul e nevoie să știe unde este localizat utilizatorul și în ce direcție se uită. În mod normal, utilizatorul explorează mediul printr-un display care reprezintă imaginea împreună cu informația augmentată.

Termenul de *urmărire*, sau *tracking* calculează *pose*-ul relativ (locație și orientare) a camerei în timp real. Aceasta reprezintă una din problemele fundamentale ale realității augmentate.

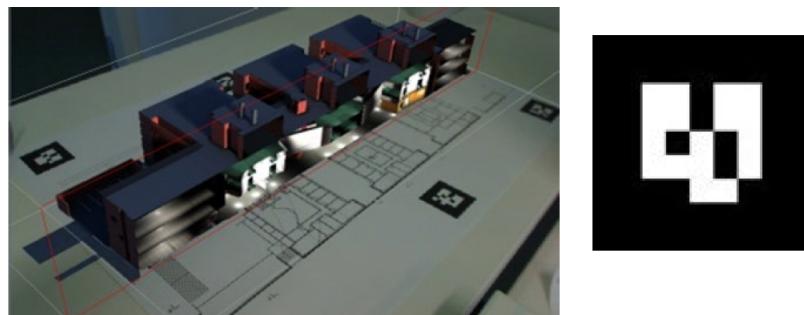


Figura 4.12: Imaginea din stânga: model augmentat amplasat asupra unui plan de clădire. Imaginea din dreapta: exemplu de marker. Din [6]

4. Fundamente Teoretice

În detecția bazată pe markere, sistemul detectează marker-ul, îl identifică și calculează *pose*-ul.

Un marker de calitate este ușor detectabil și fiabil în toate ipostazele. Diferențe în luminozitate pot fi mai ușor detectate decât diferențele cromatice (culoare) folosind tehnici de viziune computațională. Aceasta se datorează înregistrărilor incorecte: un obiect poate să-și modifice culoarea în imagine depinzând de ce mai există în câmpul vizual. Așadar variațiile de iluminare între culorile percepute contribuie la detecția mai ușoară a obiectelor.

De asemenea, sistemul ar trebui să fie capabil să calculeze *pose*-ul camerei folosind marker-ul detectat. De obicei, patru puncte sunt suficiente pentru a calcula *pose*-ul unei camere în mod unic și cea mai simplă metodă de a determina aceste puncte este un pătrat.

În consecință, multe sisteme de markere folosesc culori alb-negru la aceste pătrate. Un exemplu de astfel de marker este ilustrat în Figura 4.12 îm preună cu obiectul 3-D.

Procedura de detecție a marker-ului

Scopul principal al procesului de detecție al marker-ului este de a găsi contururile potențialelor markere și de a deduce locațiile colțurilor marker-ului din imagine. Pe lângă aceasta, sistemul de detecție trebuie să confirme sau să infirme validitatea și existența marker-ului și să îi descifreze identitatea. În final, sistemul calculează *pose*-ul folosind informația din detecția locației marker-ului. Procedura de bază este alcătuită din următorii pași:

1. Achiziția imaginii
 - achiziția intensității imaginii;
2. Pre-procesare
 - procesarea imaginii low-level;
 - rezolvarea distorsiunilor;
 - detecția muchiilor și potrivirea lor;
 - detecția colțurilor marker-ului;
3. Detectia potențialelor colțuri ale markerelor;
 - rejecția/respingerea markerelor neconcluzente;
 - test de acceptare pentru potențialele markere;
4. Identificarea și decodificarea markerelor;
 - template matching (template markers);
 - decodificare (data markers);

5. Calculul *pose*/-ului;

- estimarea *pose*-ului marker-ului;
- calcul iterativ al *pose*-ului și îmbunătățire a acurateței;

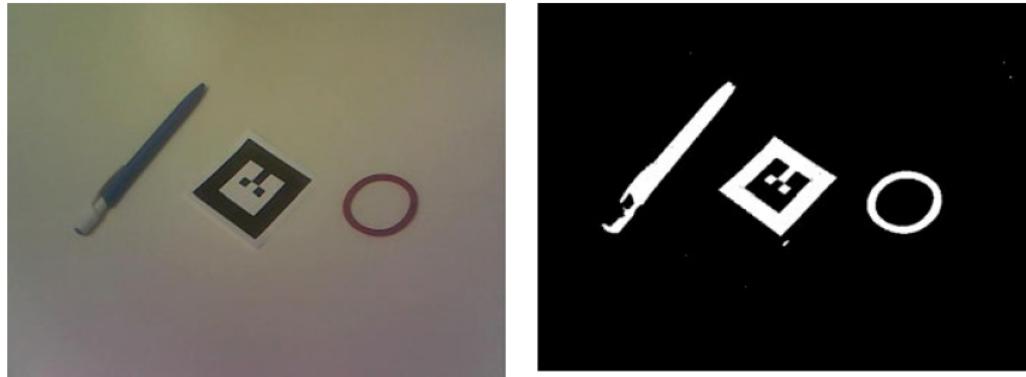


Figura 4.13: Figura din stânga: imagine originală; Figura din dreapta: imaginea după tehnica de "adaptive threshold". Din [6]

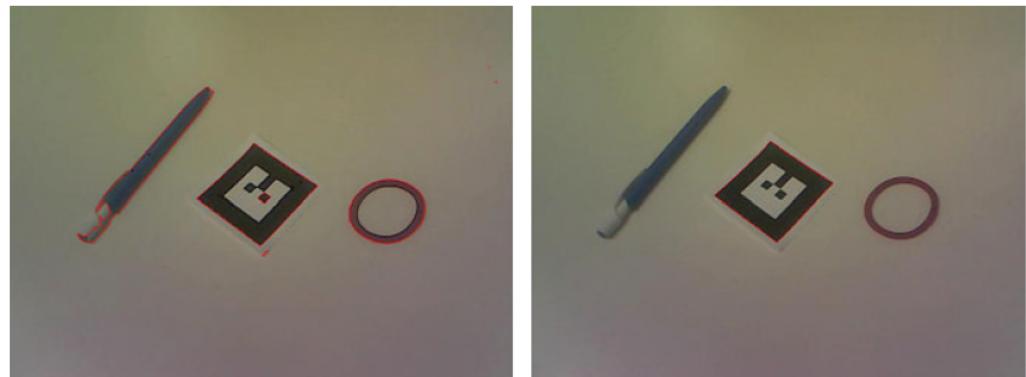


Figura 4.14: Exemplu de detecție a muchiilor. Figura din stânga: contururi detectate; Figura din dreapta: muchiile care rămân după aplicarea testului celor 4 colțuri pentru pătrate. Din [6]"

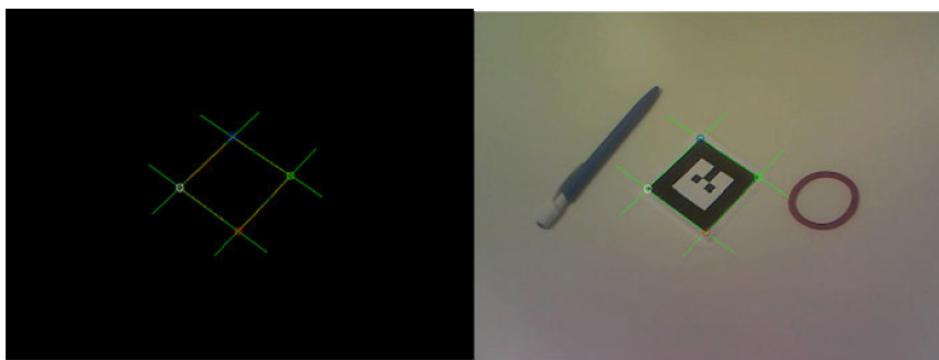


Figura 4.15: Exemplu de potrivire a muchiilor. Figura din stânga: potrivirea muchiilor în mediu fără distorsiuni; Figura din dreapta: detectia muchiilor peste imaginea originală. Din [6]

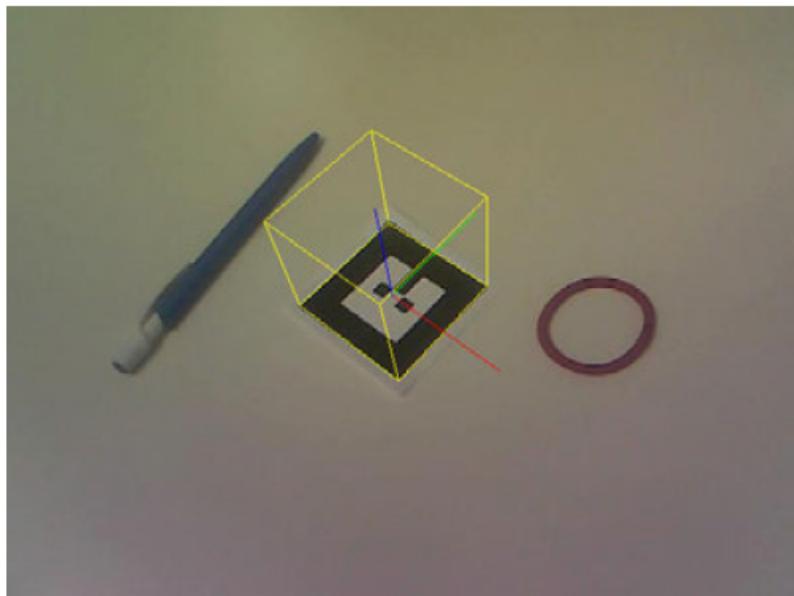


Figura 4.16: Un cub augmentat deasupra marker-ului detectat. Sistemul de coordonate (X,Y,Z) este randat cu (roșu, verde, albastru). Din [6]

Capitolul 5

Prezentarea contribuțiilor autorului: Prototipul AR-Cloud-IoT

Arhitectura prototipului propus este rezultatul combinării sinergetice între domeniile menționate în capitoalele anterioare. Sinergia aceasta constă în asocierea celor 3 mari componente: IoT, Cloud și AR pentru îndeplinirea unui scop sau funcții comune. În consecință, se disting 3 mari nivele în arhitectură: nivelul dispozitivului IoT cu senzorii atașați, nivelul serviciului de cloud, care permite stocarea datelor senzoriale de pe dispozitiv și componenta de aplicație mobilă care va primii datele stocate în cloud iar apoi le va reprezenta în contextul AR.

Capitolul 4 trecea în revistă anumite modele de comunicare. În contextul proiectului curent este abordat tipul de comunicare *Device-to-gateway* în ceea ce privește gestiunea datelor senzoriale. Întrucât la nivel de cloud, se vor folosi servicii AWS (Amazon Web Service), este destul de clar că modelul va fi unul SaaS(Software as a Service) iar la nivel de platformă, se va utiliza AWS DynamoDB. În final, partea care conține contribuția personală o constituie aplicația mobilă capabilă să ruleze pe dispozitive cu sistem de operare Android.

În Figura 5.1 este ilustrată arhitectura conceptuală la un nivel înalt. Se disting cele 3 mari modularizări:

Dacă privim aceste nivele din perspectiva unu proces complex, ele se asociază fiecare cu câte o fază:

1. Faza Achiziției de Date și Transmisie în Cloud
2. Faza Stocării în Cloud
3. Faza de Augmentare pe aplicația mobilă

Aceste faze vor fi abordate singular în capitoalele ce urmează. Este important de reținut că fiecare fază este caracterizată prin diferenți pași, diferite configurații, diferite tehnologii, diferite concepte și abordări totodată. Desigur, este foarte important de

5. Prezentarea contribuțiilor autorului: Prototipul AR-Cloud-IoT

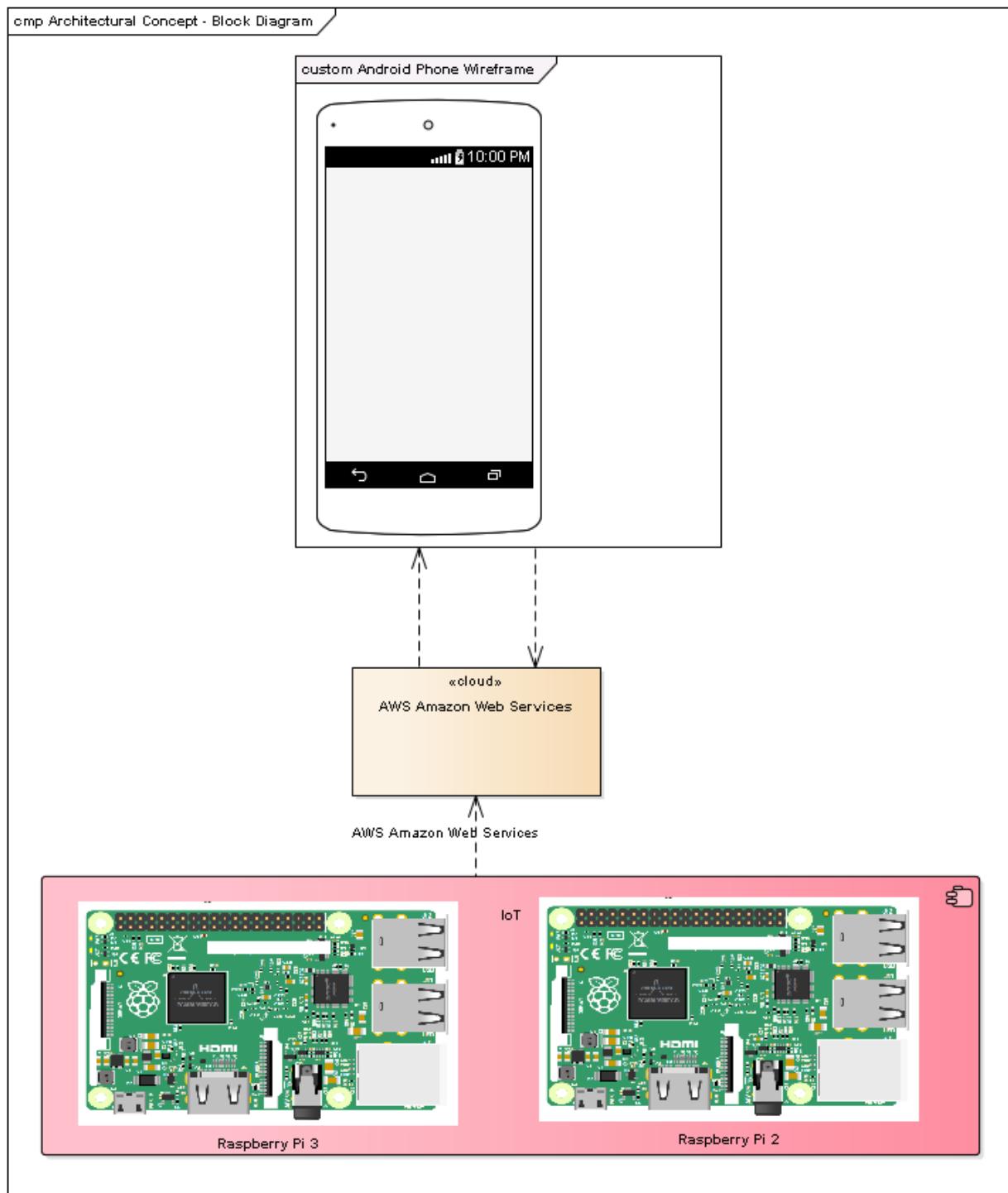


Figura 5.1: Diagrama bloc de arhitectură

menționat că proiectarea unui sistem de această manieră conduce la una din caracteristicile și specificațiile definitorii ale studiului prezent, și anume **decuplarea** și interschimbarea componentelor și a modulelor.

Diagrama "Use Case" oferă detalii relevante cu privire la sistemul sau aplicația care este în curs de dezvoltare. Modelul identifică specificațiile funcționale ale aplicației. Descrierea opțiunilor utilizatorului și a acțiunilor conduce spre un scop final bine definit. În termeni de funcționalitate internă, modelul "Use Case" nu descrie niciun fel de funcționalitate.

În modelul prezent există un singur actor-utilizator. Acesta poate deschide sau închide aplicația, scaneză ținta, interacționează cu UI-ul pentru a aduce datele în aplicație, poate vizualiza datele și poate face un snapshot ecran-ului smartphone-ului. Figura 5.2 afișează aceste acțiuni.

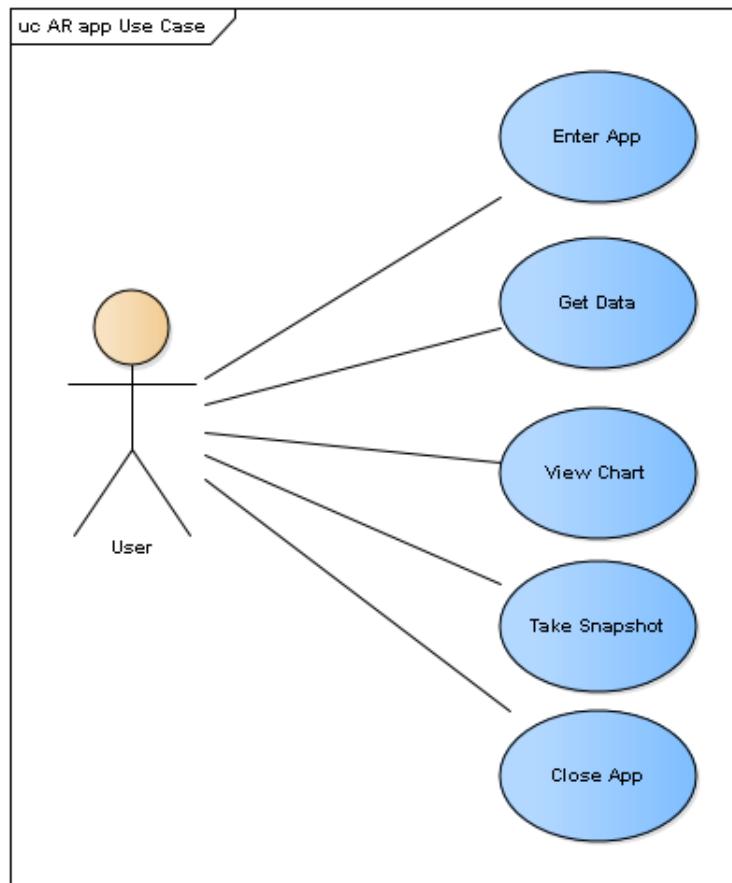


Figura 5.2: Diagrama Use Case

5.1 Proiectarea și modelarea componentelor arhitecturale

Diagrama de lansare (deployment) ilustrează topologia componentelor fizice care iau parte în sistemul de față. De asemenea, în Figura 5.3 sunt ilustrate și mediile de execuție la fiecare nivel. Prin urmare, la nivel fizic se disting următoarele dispozitive:

1. Raspberry Pi;
2. Hat-ul de senzori integrați;
3. Amazon Web Servicesca și componenta Cloud;
4. Smartphone-ul Android;

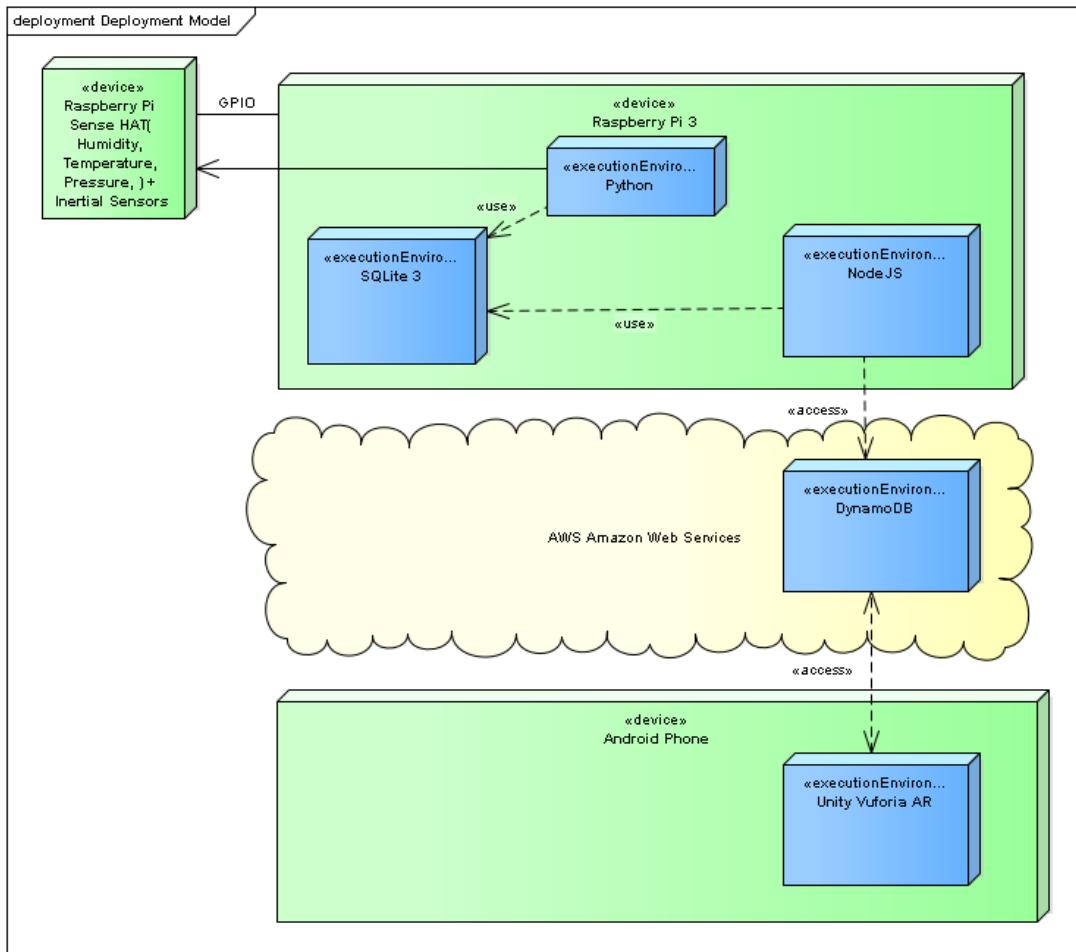


Figura 5.3: Diagrama de lansare

Datele senzoriale de tipul *Environment* (*Temperatură, umiditate, presiune, luminozitate*) și *IMU* (*Inertial Movement Unit - accelerometru, giroscop, etc.*) sunt prelucrate de hat, preluate prin intermediul microcontroller-ului Raspberry Pi iar apoi stocate într-o bază de date NoSQL de tip document, DynamoDB, care este oferită sub formă de serviciu de către platforma cloud Amazon Web Services. Aceste date sunt ulterior prelucrate de aplicația mobilă Android. Această aplicație a fost construită în motorul de joc (Unity game engine) folosind librăria de realitate augmentată Vuforia.

O scurtă trecere în revistă cuprinde următoarele componente:

- Componenta de **Colecție a Datelor**;
 - captarea valorilor senzoriale la intervale regulate de timp;
- Componenta de stocare locală a datelor într-o **Bază de Date**;
 - stocare locală pe microcontroller pentru o perioadă limitată de timp;
- Componenta **Web Server/serviciu REST**;
 - inițializarea serviciului REST;
 - configurarea conexiunii spre serviciul cloud;
 - conectarea la baza de date locală;
 - transmisia asincronă a datelor spre serviciul de baze de date din cloud;
- Componenta **serviciu Baze de Date în cloud**;
 - stochează și persistă datele în formatul corespunzător;
- Componenta de **Servicii la nivel de aplicație mobilă AR** ;
 - configurarea serviciului de cloud;
- Componenta de **Procesare a datelor din Cloud**;
 - prelucrare a datelor din cloud;
 - configurare a aplicației de realitate augmentată;
 - configurare a markerelor;
- Componenta de **Interfețe-Utilizator**;
 - reprezentare a datelor sub formă de diverse grafice;

Fiecare componentă îndeplinește o funcționalitate specifică și este asociată fie cu o funcționalitate anume, fie cu o tehnologie separată, fie cu un scop anume. Figura 5.4 ilustrează la nivel de aplicație componente care iau parte în acest sistem separate la nivel de modul. În capitolele care urmează aceste componente vor fi detaliate în mod corespunzător, în contextul fazei de care aparțin.

5. Prezentarea contribuțiilor autorului: Prototipul AR-Cloud-IoT

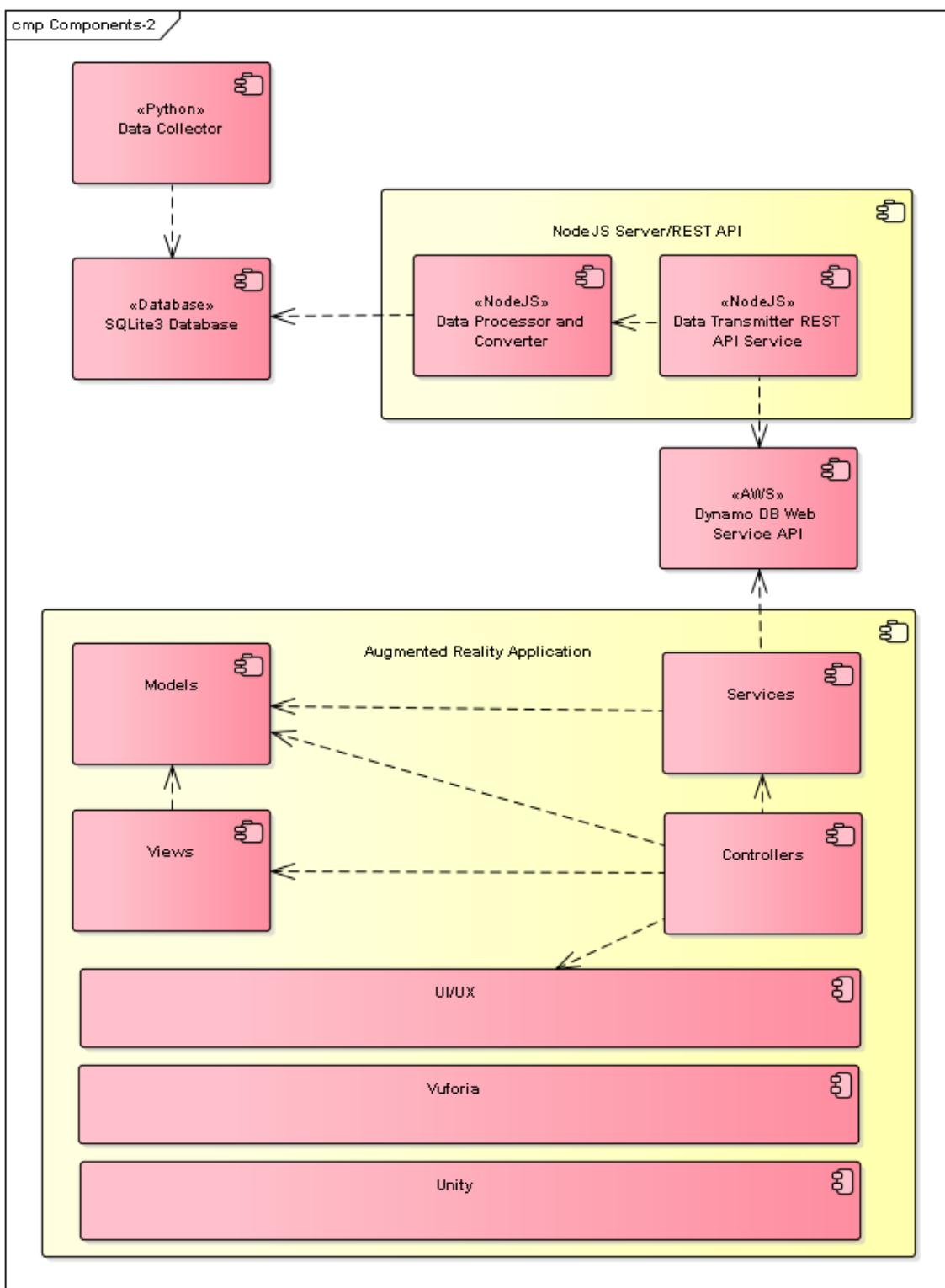


Figura 5.4: Diagrama de componentă

5.1.1 Faza I: Achiziția și transmisia datelor

Sense HAT-ul se amplasează fizic peste Raspberry Pi, peste interfața GPIO a dispozitivului. Compatibilitatea între cele 2 este garantată de producător.

Colectarea datelor

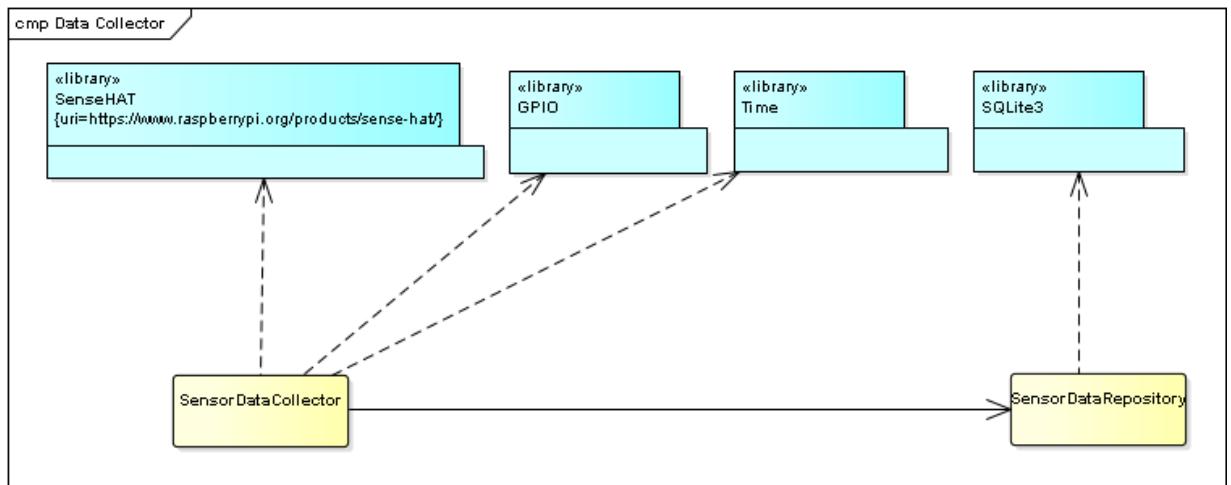


Figura 5.5: Componenta de colectare a datelor senzoriale

Această componentă are rolul de a prelua din datele senzoriale din mediu folosind placă de captură *Sense HAT*. Librăria scrisă în Python asigură compatibilitatea, flexibilitatea și implementarea într-un mod ușor. Nu există nevoie de a converti datele din format analogic în format digital. Pe de altă parte, la dispozitivul Raspberry Pi 3 folosit, s-au utilizat senzori de mediu analogici cu 3 fire unde a fost necesară convertirea digitală din semnal analogic. Diagrama din Figura 5.5 oferă o imagine de ansamblu a acestei componente.

SensorDataCollector

Este script-ul scris în Python care preia valorile senzoriale. Se utilizează librăriile de acces ale interfeței GPIO și prin intermediul librăriei SenseHAT, valorile sunt recepționate. Apoi, se stabilesc intervalele regulate la care se citește de pe senzor. În continuare, se folosește funcționalitatea de pe Sensor Data Repository pentru a salva valorile în baza de date.

SensorDataRepository

Acest script reprezintă accesul la baza de date SQLite3. Prin acesta se asigură conexiunea la baza de date și se expun operațiile de CRUD (create, retrieve, update, delete) pentru SensorDataCollector. Desigur, pentru cazul acesta particular, se utilizează doar operația de inserare.

Baza de Date SQLite3

Baza de date SQLite are rolul de a stoca valorile de pe senzor. Este foarte ușor de menținut întrucât este un tip de baze date specific dispozitivelor mobile. În Figura 5.6 este expus modelul de date.

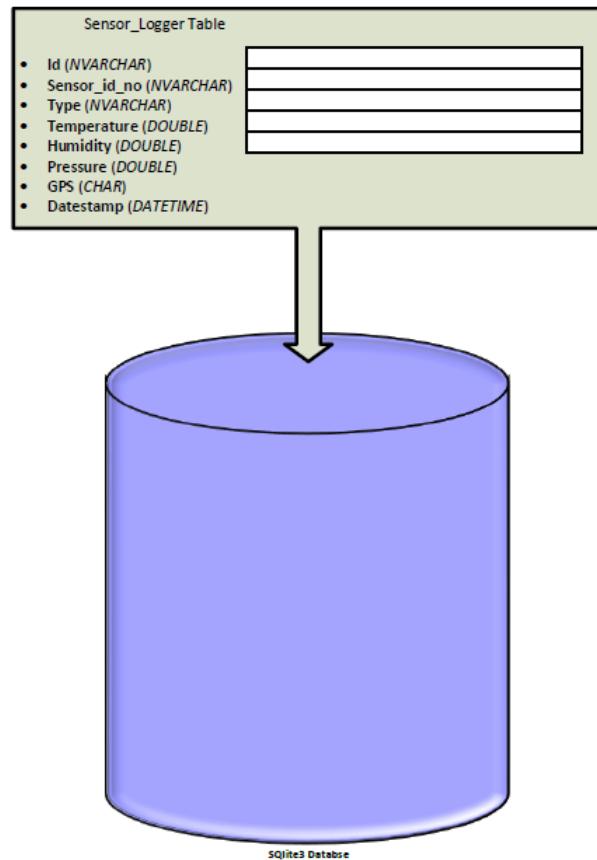


Figura 5.6: Modelul de date.

În Figura 5.7 se ilustrează organograma achiziției datelor și activitățile care iau parte la aceasta.

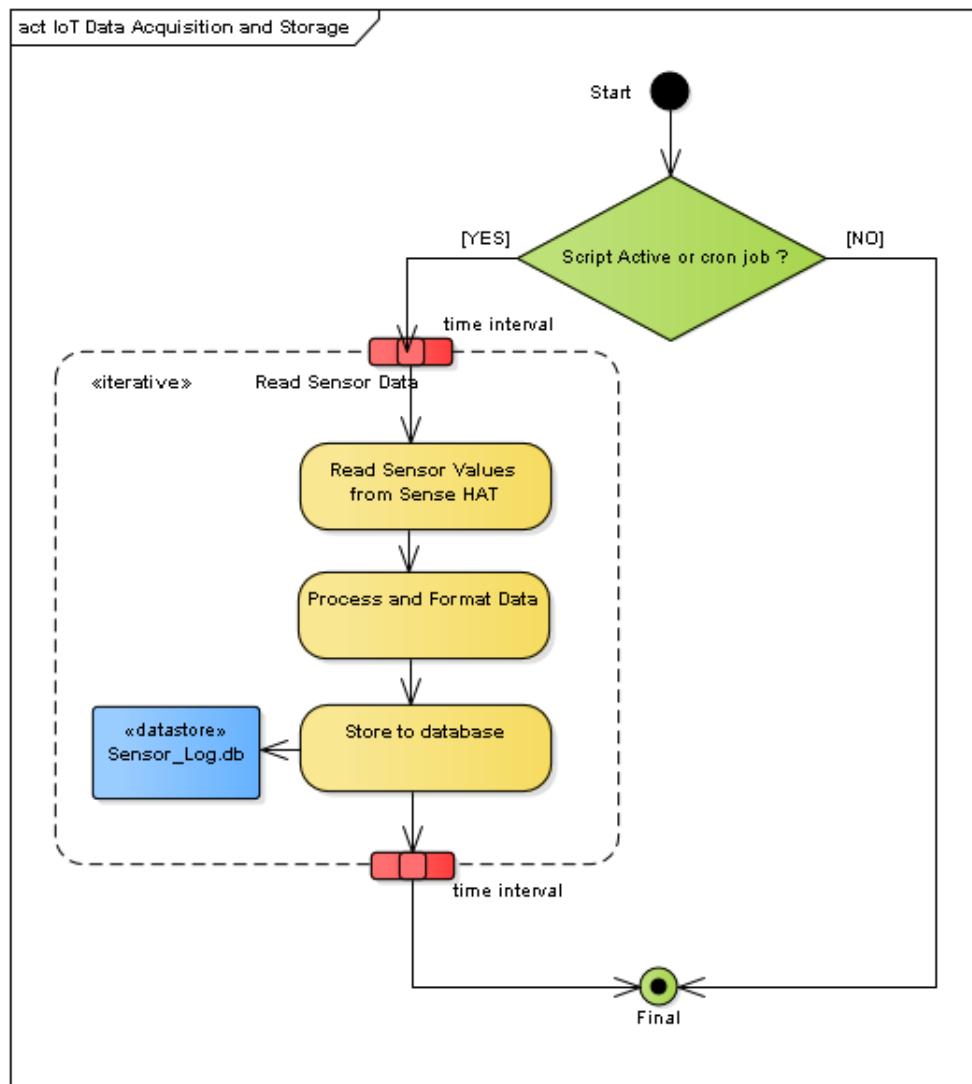


Figura 5.7: Organigrama achiziției de date.

Componenta Web Server/ Serviciul REST

Serviciul REST (poate fi numit și web server întrucât rulează cod JavaScript server-side) are rolul esențial de a prelua datele din baza de date locală, de a parsa, pregăti și formata datele în formatul JSON. De asemenea, asigură autorizarea serviciului de cloud, stabilirea end-point-ului și a regiunii, precum și pregătirea pentru transmisia în cloud. În final, pasează datele în serviciul de cloud. Componentele care iau parte la acest proces sunt ilustrate în Figura 5.8 .

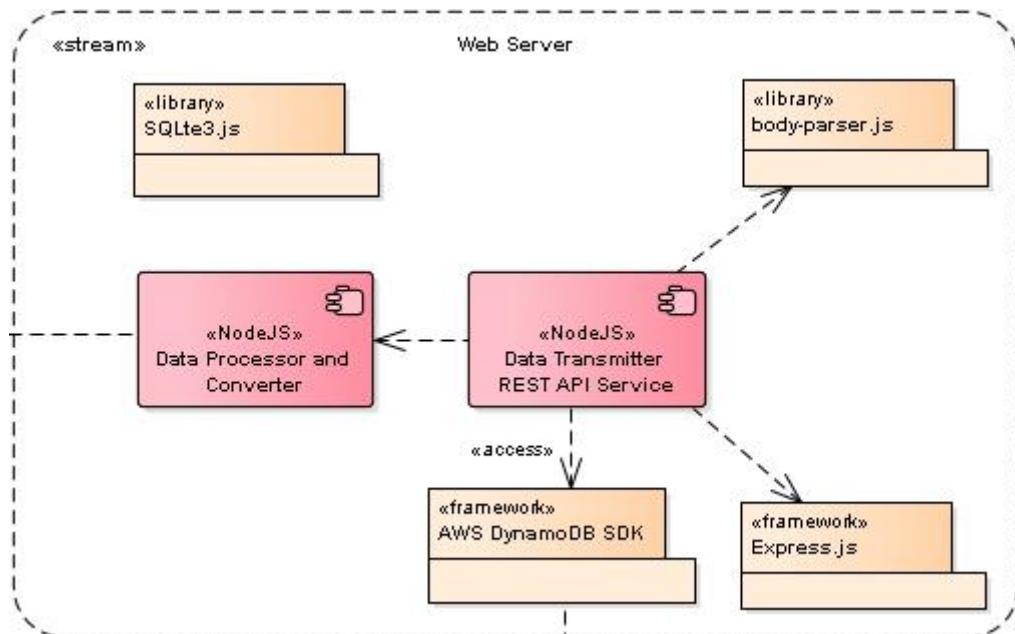


Figura 5.8: Web-server/Serviciul REST

Data Transmitter REST Service

Acest script are rolul de a facilita transmisia valorilor în baza de date din Cloud. Prima dată se autorizează clientul, se stabilește regiunea endpoint-ul. În continuare, se realizează conexiunea la baza de date, de unde se preiau datele. După aceasta, se crează instanța serviciului, fiecare valoare fiind convertită în formatul corespunzător, i.e. JSON urmând ca, în final, datele să fie transmise asincron spre cloud. Acești pași pot fi observați și în organigramă din Figura 5.9.

Librăriile și framework-urile dependente sunt următoarele: SQLiteJS, ExpressJS, body-parser.js și AWS DynamoDB pentru JavaScript pe NodeJs.

De asemenea, există script-uri adiționale pentru diferite operații secundare, cum ar fi crearea tabelului. Există de asemenea o componentă care nu este ilustrată în diagrame, însă se poate regăsi în secțiunea de Appendix. Aceasta reprezintă un REST API care expune toate valorile din baza de date înspre exterior pentru a fi consumate (format JSON).

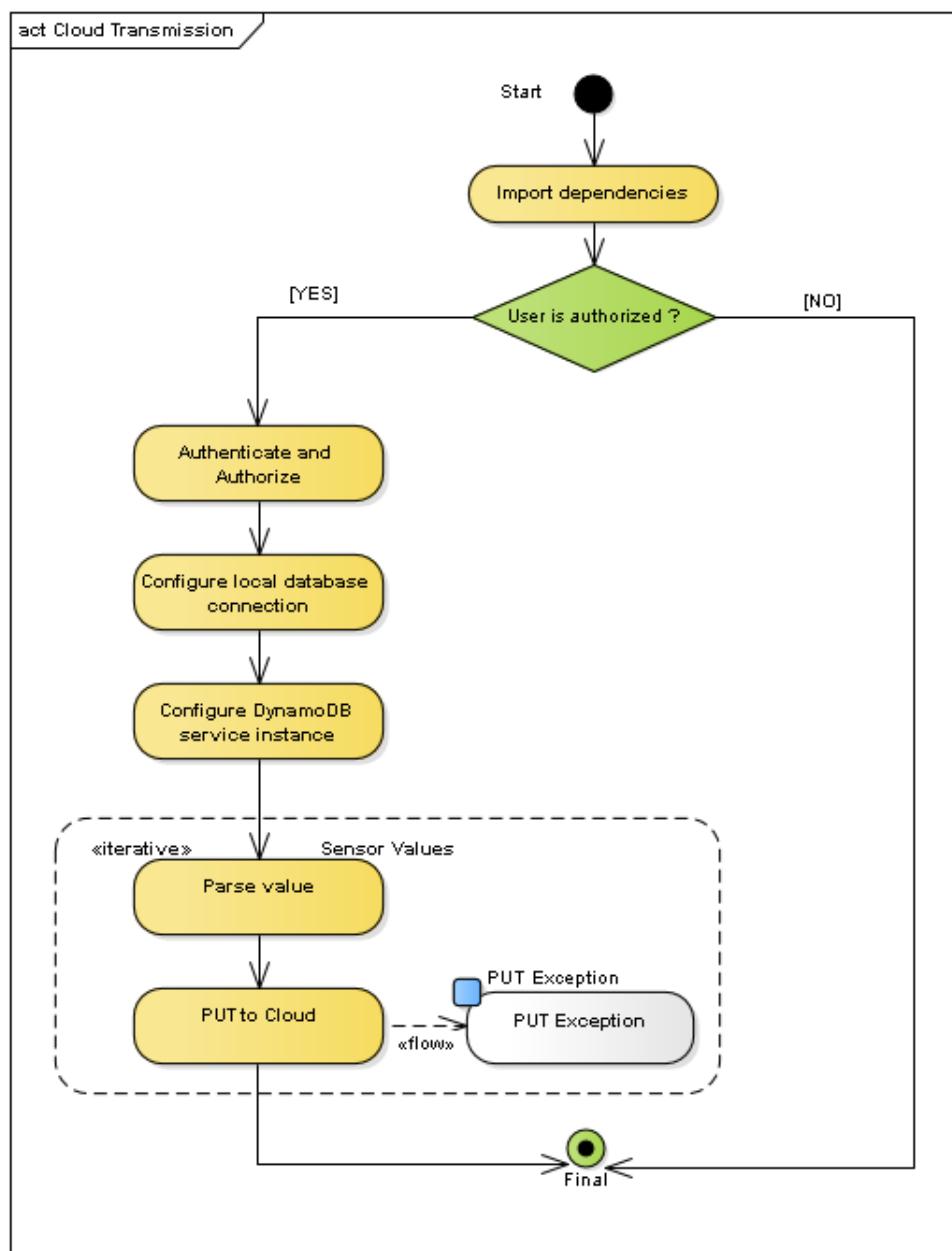


Figura 5.9: Organigrama transmisie de date.

5. Prezentarea contribuțiilor autorului: Prototipul AR-Cloud-IoT

Diagrama din Figura 5.10 rezumă în întregime această fază a achiziției de date și mapează de asemenea tehnologiile și limbajele de programare principale. În sectiunile viitoare aceste tehnologii vor fi prezentate în detaliu.

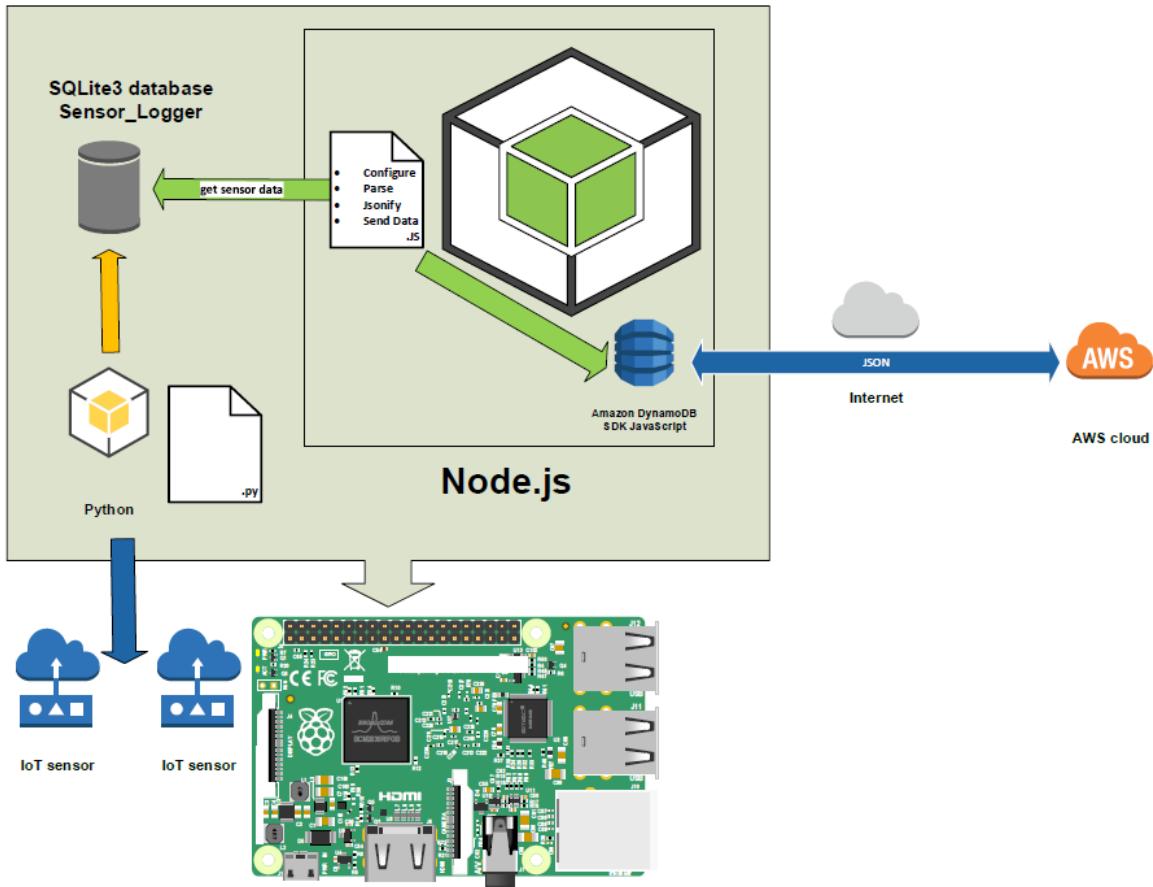


Figura 5.10: Achiziția și transmisia datelor senzoriale.

5.1.2 Faza II: Stocarea in Cloud

Serviciul de stocare în cloud este bazat pe o paradigma diferită față de bazele de date relaționale. De asemenea, la crearea tabelei/tabelelor de baze de date este nevoie să se specifice anumite caracteristici. Pentru a crea un tabel de baze de date în tehnologia DynamoDB, pe lângă nume, trei alte caracteristici sunt esențiale:

Schema cheilor (Key Schema)

Este nevoie să se specifice tipul cheii (de obicei se specifică o cheie de partizionare Partition Key) și o cheie de sortare (Sort key).

Definirea atributelor (Attribute Definitions)

5.1. Proiectarea și modelarea componentelor arhitecturale

Apoi, este necesar ca aceste două chei să fie definite în ceea ce privește atributele, numele și tipul. În consecință, toate atributele pe lângă cele două menționate, vor fi adăugate, neînținându-se cont de numărul de attribute - schema variabilă.

Gestionarea capacitatei per unitate (Provisioned Throughput)

În ultimul rând este necesară specificarea numărului de unități de citire și scriere. Figura 5.11 prezintă vizual schematizarea la nivel de arhitectură a acestei baze de date.

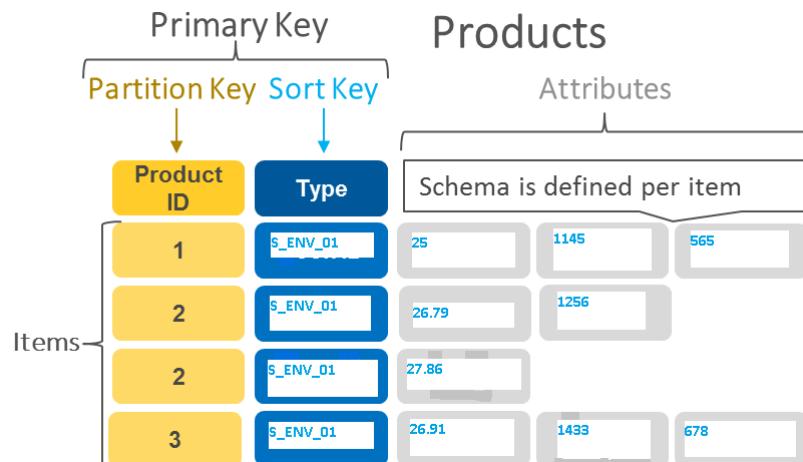


Figura 5.11: Exemplu DynamoDB cu chei de partitie și sortare.

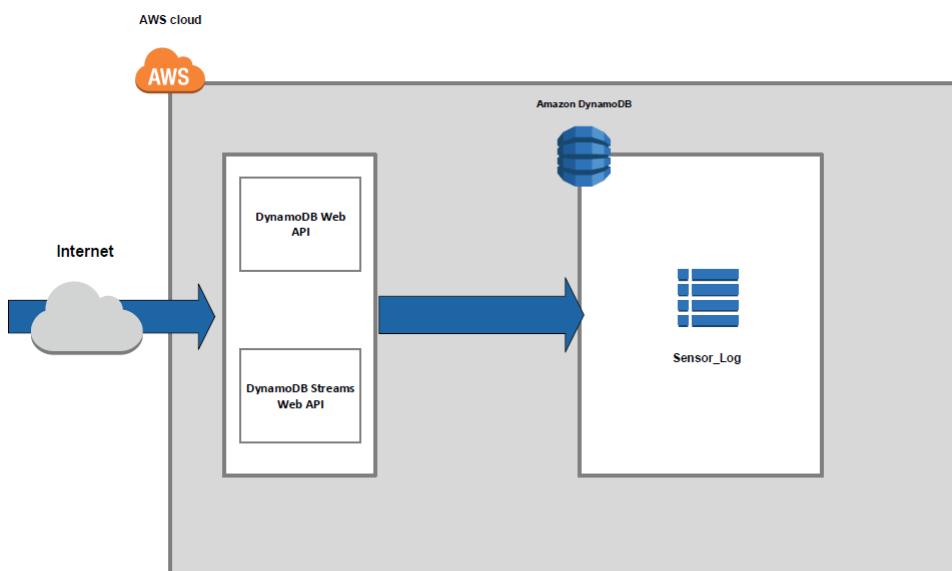


Figura 5.12: Stocarea datelor provenite de pe dispozitivul IoT.

Figura 5.12 ilustrează pe scurt cea de-a doua fază a procesului sistemului.

5.1.3 Faza III: Aplicația mobilă de Realitate Augmentată

Aceasta reprezintă ultima fază din procesul sistemului. Datele sunt achiziționate în momentul în care utilizatorul decide asta, iar reprezentarea augmentată a datelor sub formă de grafic rămâne de asemenea la latitudinea utilizatorului. Aplicația este construită în totalitate utilizând motorul de joc Unity. O prezentare și descriere mai amănunțită a acestei platforme o regăsim în [55]. Pentru componenta de realitate augmentată, librăria Vuforia a fost integrată cu Unity. Pentru comunicarea cu serviciul cloud a fost necesară configurarea pachetului AWS DynamoDB pentru Unity/.NET. În cele ce urmează vor fi analizați pașii importanți în expunerea celor mai importante fluxuri ale informației cât și definirea modulelor, componentelor, librăriilor și claselor esențiale.

Figura 5.13 reprezintă diagrama bloc a acestei faze. Se pot distinge cele două mari componente: componenta de Cloud și cea de dispozitive mobile. Diagrama cuprinde de asemenea mijloacele de autentificare și autorizare în serviciul de Cloud prin Amazon IAM și Cognito.

Figura 5.14 reprezintă structura aplicației mobile în ceea ce privește tehnologiile utilizate și modul în care datele care vin din exterior sunt gestionate și prelucrate.

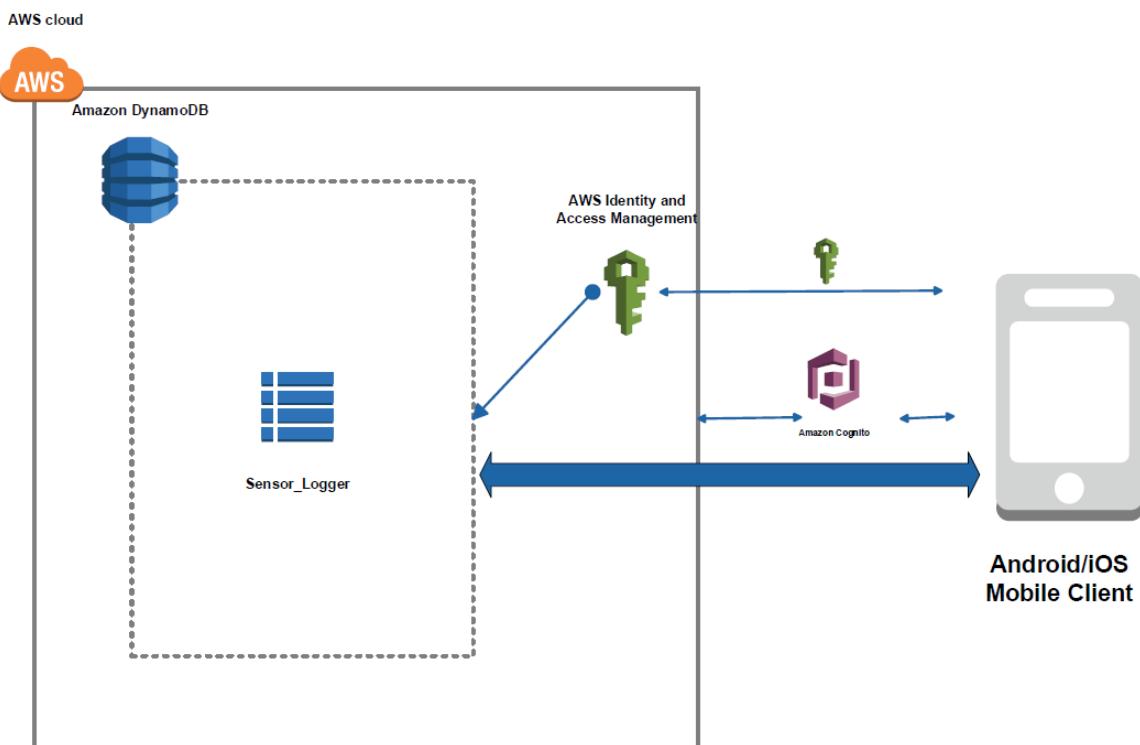


Figura 5.13: Achiziționarea datelor din cloud pe aplicația mobilă

Următoarele componente sunt implicate în această fază:

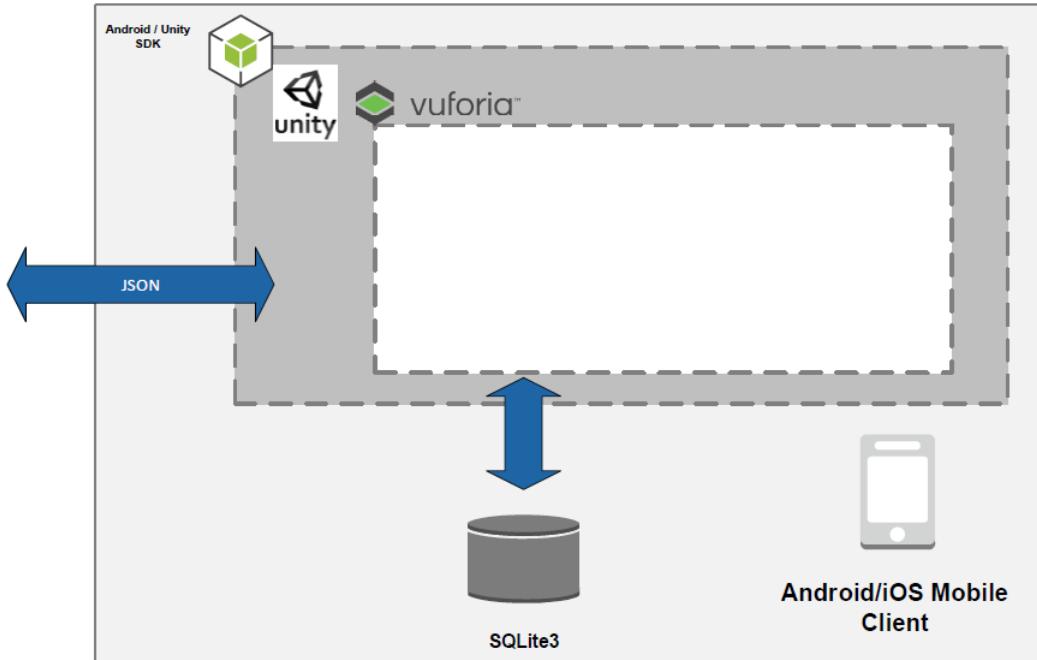


Figura 5.14: Structura aplicației mobilă

Componenta Unity

Această componentă oferă suportul, infrastructura, librăriile, mediul de dezvoltare și plugin-urile necesare dezvoltării aplicației.

Componenta Unity

Componenta Vuforia reprezintă componentă care facilitează integrarea realității virtuale cu motorul Unity. Integrează trackerele generate și implementează algoritmii necesari pentru procesarea datelor.

Componenta UI/UX

Această componentă integrează controalele și elementele de interfață Android cu libraria Vuforia și conține următoarele librării sau plugin-uri:

1. Chart and Graph - este utilizată pentru a reprezenta datele sub forma unui grafic tri-dimensional;
2. Android Ultimate - este utilizată pentru a integra anumite funcționalități în aplicația Android;

Componenta Views

Componenta aceasta conține scenele relevante pentru rularea aplicației. Obiectele 3D sunt atașate sub forma unu game object. Fiecare game object poate îngloba unul sau mai multe game objects la rândul lui într-o structură ierarhică. [55] prezintă în detaliu

5. Prezentarea contribuțiilor autorului: Prototipul AR-Cloud-IoT

modul de lucru în Unity și cele mai importante concepte. Aceste scene conțin atașate script-urile de control care conțin logica.

Componenta Controllers

Controller-ele sunt asociate cu o scenă anume și conțin logica de achiziție a datelor și de reprezentare a obiectelor 3D. Accesează repository-urile folosind serviciul de AWS și folosesc clasele de la nivelul de UI/UX specifice.

Componenta Services

Componenta de servicii înglobează serviciile AWS DynamoDB și serviciul de Repository pentru senzori. Este accesată de componenta de control pentru a stabili o configurare a serviciului și pentru a efectua apelurile REST către Cloud.

Componenta Models

Componenta conține clase-container simple care mapează fiecare proprietate a unei valori senzoriale. Este pasată și utilizată la nivel de controller pentru o conversie corectă a datelor JSON în formate corespunzătoare aplicației mobile.

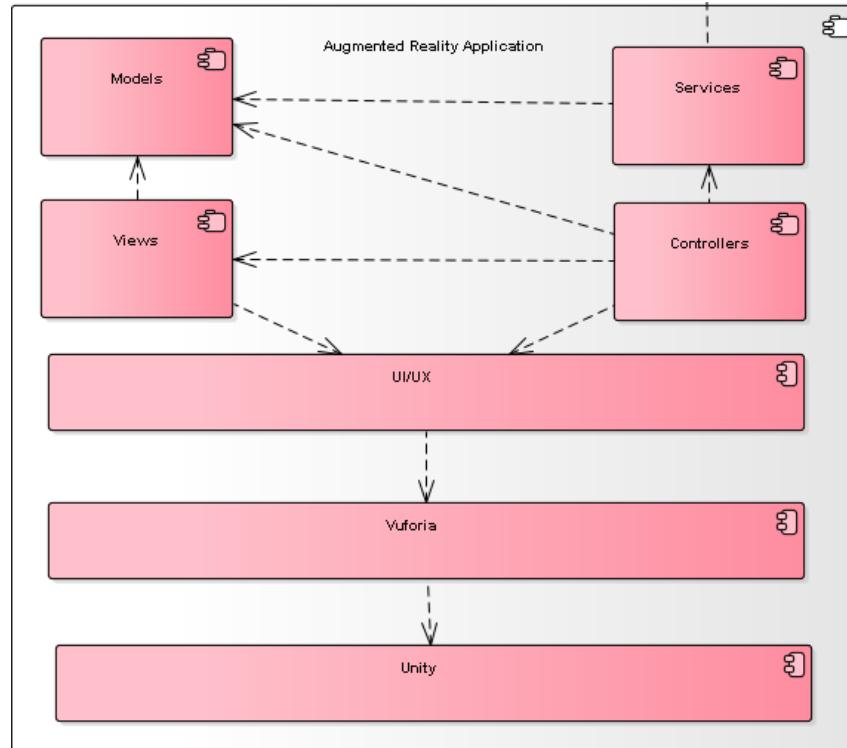


Figura 5.15: Componentele aplicației

Diagrama de clase

Diagrama de clase reprezintă o soluție pentru specificațiile inițiale și pentru entitățile identificate în faza de analiză a soluției. Deși componentele asociate fiecărei clase nu sunt

foarte vizibile, ele pot fi deduse din responsabilitatea fiecărei clase. Proiectarea modulară bazată pe soluții enterprise este extrem de eficientă pentru crearea de aplicații software scalabile.

Diagrama din Figura 5.16 surprinde clasele cele mai importante în procesul de achiziție a datelor senzoriale de temperatură și reprezentare a lor. Prin urmare, se disting foarte ușor clasele specifice serviciului REST și apelurile făcute spre cloud, clasele care oferă suport pentru gestionarea datelor odată ajunse în aplicația mobilă și clasele care oferă suport în reprezentarea obiectelor în mediul virtual augmentat.

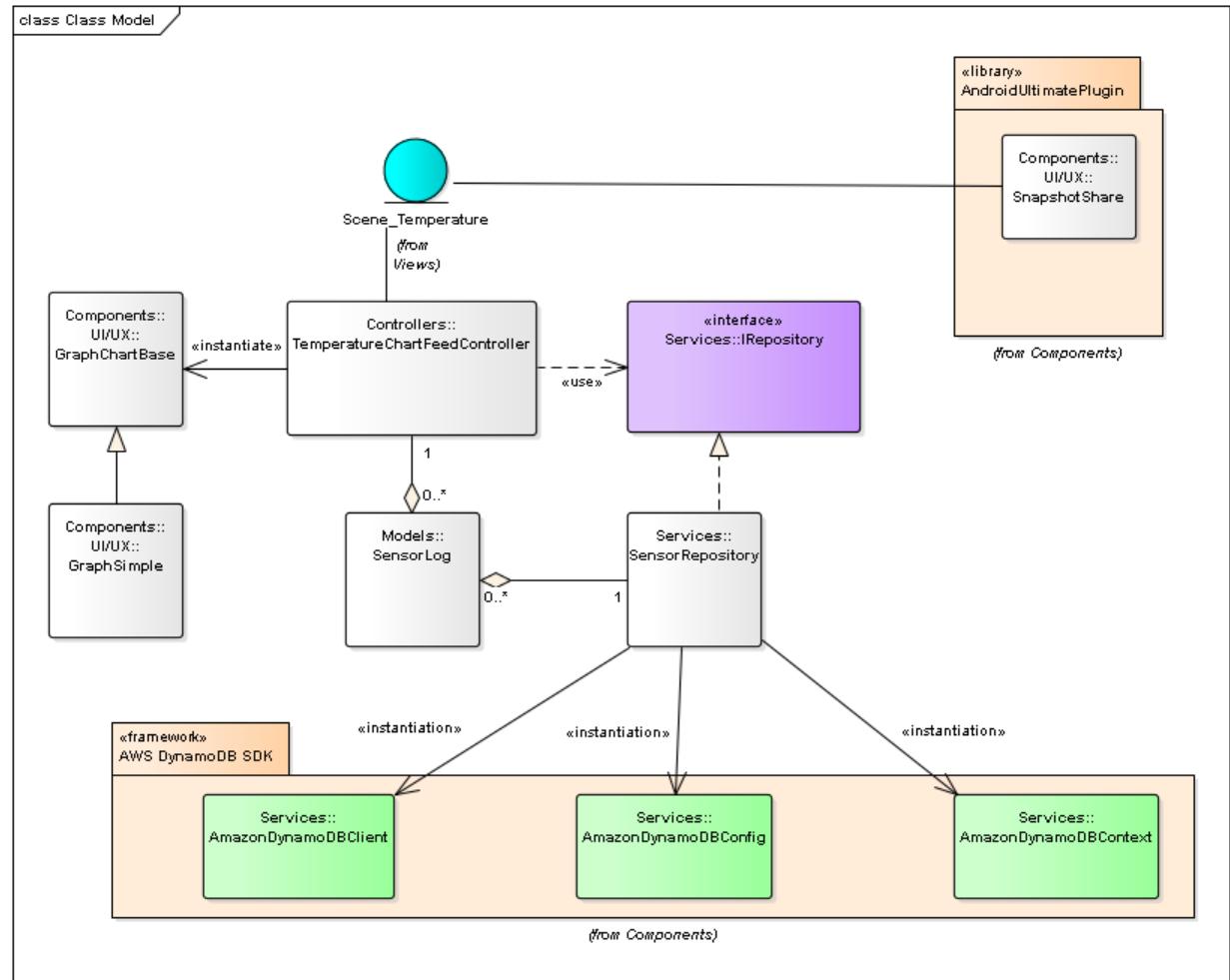


Figura 5.16: Diagrama de clase

Diagramele de secvențe

Diagramele de secvențe afișează modul în care obiectele care iau parte la proces interacționează unul cu altul în mod secvențial. Au fost identificate la acest nivel 2 diagrame importante prezentate mai jos.

Configurarea Servicului Amazon DynamodDB

Este extrem de important de menționat cele 4 clase care trebuie apelate pentru a configura serviciul REST de achiziție din cloud. Acestea sunt: Amazon.UnityInitializer, AmazonDynamoDBClient, AmazonDynamoDBConfig și AmazonDynamoDBContext.

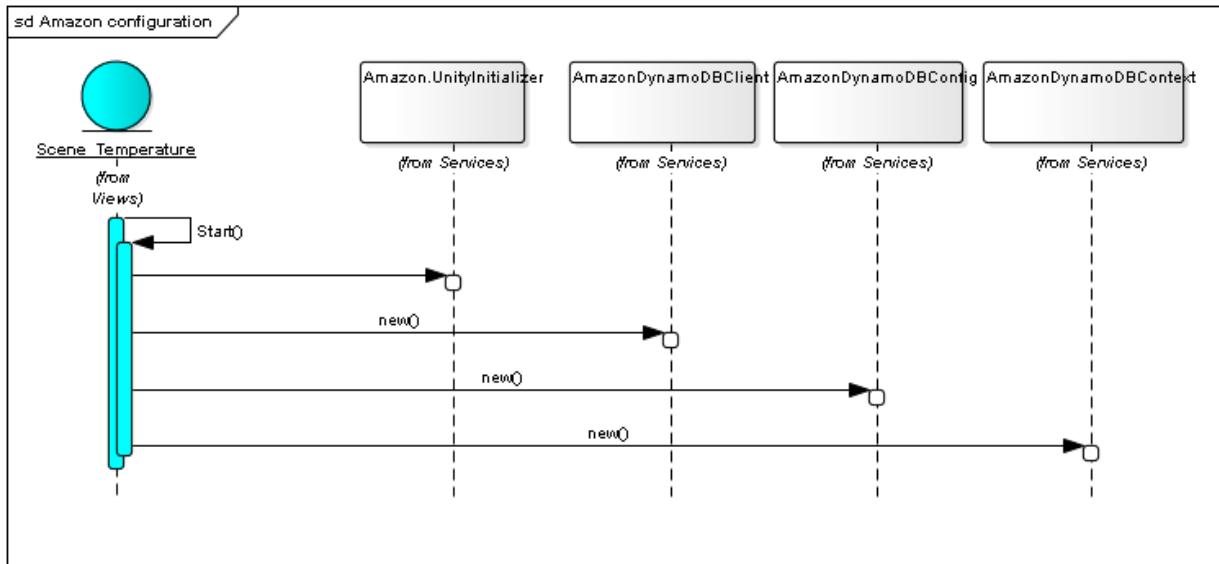


Figura 5.17: Diagrama de clase

Returnarea datelor din cloud

Returnarea datelor din cloud implică acțiunea directă a utilizatorului de a aplica comanda respectivă. Serviciul DynamoDB achiziționează asincron datele în format JSON. Aceste date sunt parsate apoi și reprezentate folosind tipuri diferite de "charts" sub formă de serie numerică și temporală. Valoarea temperaturii este afișată în funcție de data preluării acesteia. Figura 5.18 reprezintă diagrama de secvențe asociată acestui proces.

5.1. Proiectarea și modelarea componentelor arhitecturale

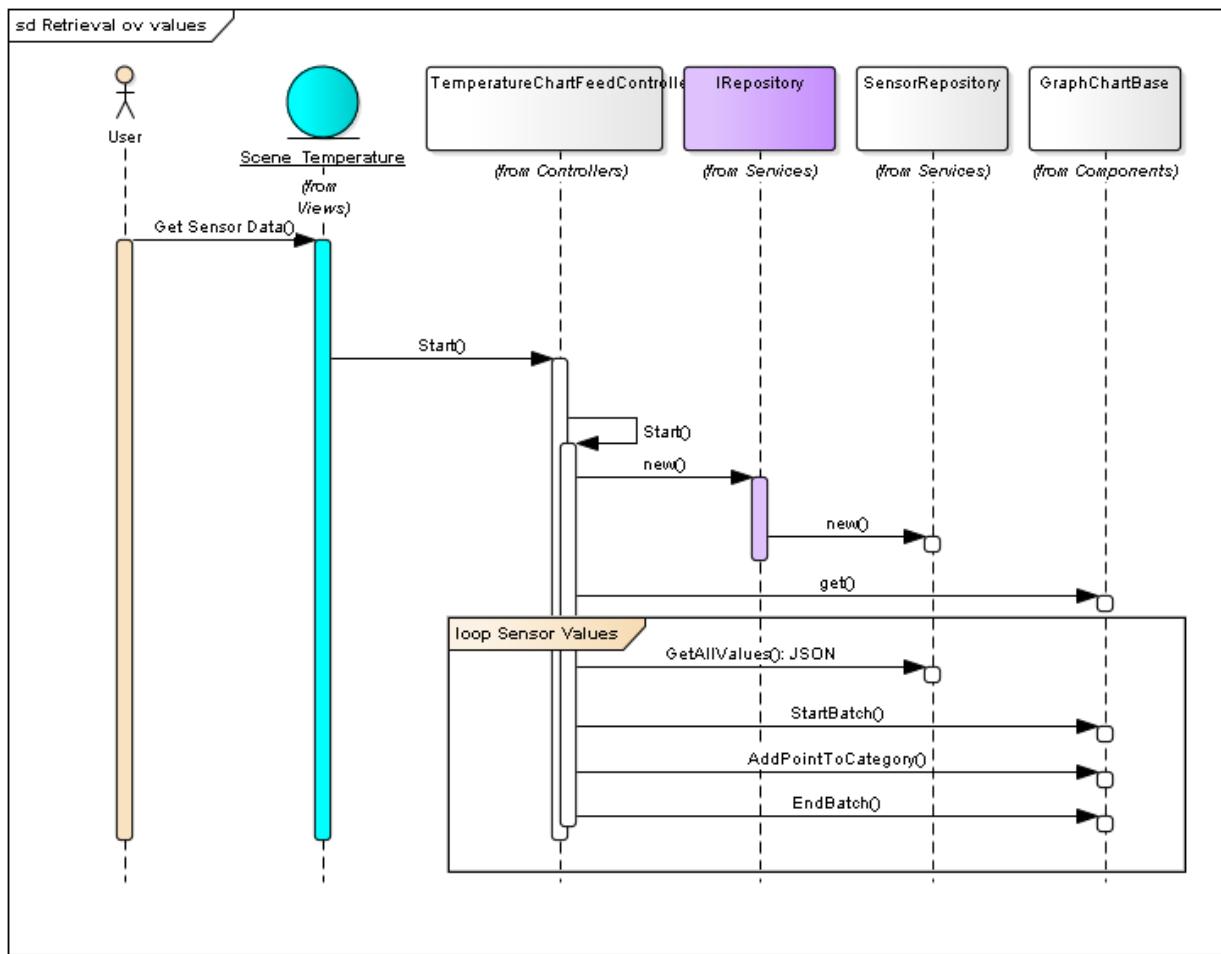


Figura 5.18: Diagrama de clase

5.2 Tehnologiile implementate

Această secțiune cuprinde tehnologiile principale pentru dezvoltarea prototipului AR-Cloud-IoT. Figura 5.19 reprezintă stiva tehnologilor utilizate.



Figura 5.19: Stiva de tehnologii

Pentru implementarea componentei IoT - achiziție și transmisie de date au fost folosite următoarele dispozitive, tehnologii, limbi de operare, framework-uri și limbi:

- Raspberry Pi - Este un SBC(single-board-computer) de dimensiunile unui card de identificare, produs în Marea Britanie și folosit în mediile academice. Inițial a avut rolul de a învăța studenții bazele științei calculatoarelor. Ulterior, s-a dovedit că este extrem de flexibil și a avut o expansiune în lumea robotică.
- Raspbian OS - Este un sistem de operare optimizat pentru familia de micro-calculatoare Raspberry Pi, bazat pe distribuția Linux numită Debian. În momentul de față Raspbian folosește mediul desktop PIXEL.

- Python - Limbaj de programare high-level extrem de flexibil și actual, care suportă multiple paradigmă de programare, inclusiv programarea orientată pe obiecte, programarea imperativă, funcțională și procedurală. Acest limbaj rulează pe majoritatea sistemelor de operare și este open-source. Este extrem de renomat în cercurile academice.
- SQLite - Sistem de gestiune a bazelor de date relaționale care nu rulează în cadrul unui server de baze de date, ci este înglobat în aplicația care îl utilizează. Se supune set-ului de proprietăți ACID. Este opțiunea preferată în stocarea locală a web-browserelor și sistemelor "embedded" (cum ar fi telefoanele mobile). Poate fi utilizat cu orice limbaj de programare.
- Node.js - Este un mediu de rulare open-source care execută cod JavaScript server-side. De asemenea, rulează script-uri pentru a produce conținut dinamic pe pagini web înainte ca pagina să fie trimisă spre browser-ul client-ului. Node.js a devenit unul dintre elementele de bază ale paradigmii "JavaScript Everywhere". Este o arhitectură bazată pe evenimente și capabilă de I/O asincron. Este optimizată scalabilitatea aplicațiilor web care efectuează des acest tip de operații dar și aplicații în timp real.
- Express.js - Framework pentru aplicații web, open-source, destinat pentru dezvoltarea de aplicații și API-uri web. Este standardul *de facto* pentru construcția de servere în Node.js.
- DynamoDB - Bază de date dezvoltată de Amazon ca și parte a AWS (Amazon Web Services). DynamoDB a fost conceput de Amazon pentru durabilitate și scalabilitate având posibilitatea să se replice pe mai multe data-centre, în funcție de preferința utilizatorului. Diferă față de alte servicii de baze de date prin faptul că utilizatorul plătește în funcție de "throughput" și nu în funcție de spațiul de stocare. Se utilizează memorii SSD extrem de rapide iar o consecință a acestora o reprezintă performanță predictibilă. Se poate integra cu Hadoop și Elastic MapReduce.

La componenta Cloud amintim serviciul de cloud Amazon Web Service oferit de Amazon.

- Amazon Web Services este o platformă cloud on-demand bazată pe o subsecție platită cu o opțiune de tehnologii gratuite pentru 12 luni. Se oferă prin aceasta un cluster de calculatoare, disponibile tot timpul prin internet. Consola AWS este virtualizată pentru fiecare utilizator în parte iar conectarea se poate face prin browser-ul acestuia. În 2016 AWS oferea mai mult de 70 de servicii inclusiv stocare, sisteme de calcul, baze de date, servere de aplicații, rețelistică, analytics, unele de dezvoltare pentru mobile și Internet of Things. Amazon Web Services oferă accesul la API-uri prin HTTP folosind servicii REST și protocolul SOAP.

Componenta de aplicație mobilă este alcătuită din următoarele tehnologii și limbaje:

5. Prezentarea contribuțiilor autorului: Prototipul AR-Cloud-IoT

- Unity - Este un motor de joc dezvoltat de Unity Technologies cu scopul principal de a dezvolta jocuri pe calculator și simulări pentru calculatoare, console și dispozitive mobile. Din 2005 până acum, Unity a extins numărul de platforme țintă la 27. Oferă suport pentru grafică 2D și 3D, funcționalitate drag-and-drop și suportă C# ca limbaj de programare/scripting.
- Vuforia - Este o librărie de dezvoltare pentru realitate augmentată pentru dispozitive mobile. Se folosește de Computer Vision pentru a detecta și urmări imagini planare și obiecte simple 3D în timp real. Înregistrarea imaginilor permite dezvoltatorilor să orienteze obiectele virtuale în corelație cu imaginile din lumea reală când acestea sunt vizualizate prin camera unui dispozitiv mobil. Vuforia oferă suport pentru limbaje de programare cum ar fi C# și de asemenea oferă o extensie pentru motorul de joc Unity.
- Android - Sistem de operare dezvoltat de Google, bazat pe kernel-ul Linux cu țintă spre dispozitivele smartphone și tablete.
- CSharp - Limbaj de programare multi-paradigmă dezvoltat de Microsoft sub inițiativa .NET

Capitolul 6

Rezultate teoretice și experimentale

Pentru testarea și validarea soluției este nevoie de efectuarea unor serii de experimente și teste. Aceste teste au fost proiectate pentru a valida arhitectura prototipului în termeni de anumite specificații non-funcționale. Aceste specificații non-funcționale se referă la performanță (timpi de acces/răspuns ai transmisiei în Cloud, achiziționarea datelor, procesarea și reprezentarea lor), scalabilitate, robustețe, fiabilitate, utilizabilitate, portabilitate, și siguranță. De asemenea în secțiunile 6.2, 6.3 și 6.4 se vor prezenta 3 scenarii din lumea reală unde ar avea sens implementarea unei astfel de soluții.

6.1 Specificații Non-Funcționale

6.1.1 Performanță

Performanță reprezintă una dintre caracteristicile definitorii pe care orice sistem, serviciu sau aplicație modernă este nevoit să o satisfacă și reprezintă una dintre caracteristicile fundamentale care au contribuit la elaborarea acestor teste. Având acest lucru în vedere, implementarea sinergetică a celor 3 componente reprezintă întradevăr o provocare din punctul acesta de vedere. Este lesne de înțeles faptul că separarea și modularizarea atâtător componente și tehnologii ar putea induce riscul de a apărea diferite "bottlenecks" care ar încetini în mod logic funcționarea prototipului. De asemenea, introducerea componentei de realitate augmentată prin librăria Vuforia necesită timp pentru a detecta și urmări markerele (rularea algoritmilor de Computer Vision) iar apoi combinarea imaginii reale cu cea augmentată. Execuția acestor algoritmi bineînțeles că va necesita un timp mai îndelungat.

În ceea ce privește tehnologiile utilizate, este evident că s-au utilizat tehnologiile actuale de ultimă generație la fiecare nivel. De la utilizarea celui mai modern microcontroller - Raspberry Pi, la rularea serviciul REST NodeJS, la accesul în Amazon Web Services și stocarea în DynamoDB până la Unity cu Vuforia și Android.

În termeni de timp de acces și răspuns, se vor evoluă situații contextuale: situația în care aplicația rulează în cadrul rețelei locale și situația în care datele vor fi stocate prin

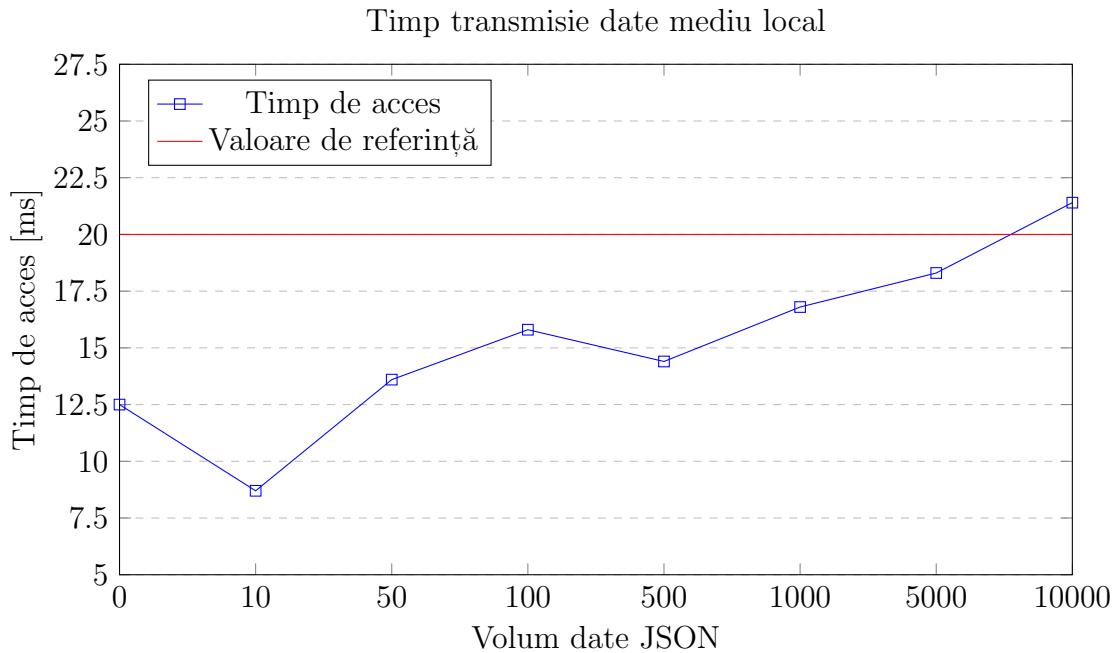


Figura 6.1: Stocarea datelor în contextul consolei locale

consola AWS.

Figura 6.1 reprezintă prima din situațiile mai sus menționate. Deși în documentația AWS DynamoDB este scris că timpii de răspuns sunt extrem de mici, sub 10 milisecunde, în practică intervin anumiți factori externi care contribuie la creșterea acestora. "Single digit millisecond" este motto-ul bazelor de date DynamoDB, însă latența introdusă poate fi extrem de ușor justificată. Se poate observa din grafic ca volumul de date nu este un factor definitiv care să reprezinte un impediment pentru aplicație. O altă explicație este faptul că pentru a menține viteze extrem de mari, DynamoDB limitează dimensiunea bloc-urilor de date și nu transmite *item*-uri mai mari de 4KB, ci le segmentează și le concatenează când valorile ajung în serviciul de baze de date.

Ceea de-a doua situație, cea în care se utilizează consola AWS introduce valori de timp adiționale din pricina rulării consolei AWS CLI la început. Această valoare e în jur de 200 de ms. Când se execută o comandă, comanda de retrieve trebuie încărcată în memorie, aduse credențialele, stabilirea unei conexiuni securizată la DynamoDB. După ce acestea sunt realizate, comanda este executată și rezultatul este afișat. Altfel, media valorilor este între 15-25 ms ceea ce este destul de aproape de valoarea definită de Amazon. O altă explicație ține de viteza luminii. Cea mai bună performanță se obține când se utilizează o instanță localizată în aceeași regiune. Este normal să existe latențe când se accesează serviciul DynamoDB de pe laptop. În continuare, există costuri majore în funcție de rețea, hardware, conexiunea la Internet și limbajele de programare folosite. Rezultatele pot fi observate în Figura 6.2.

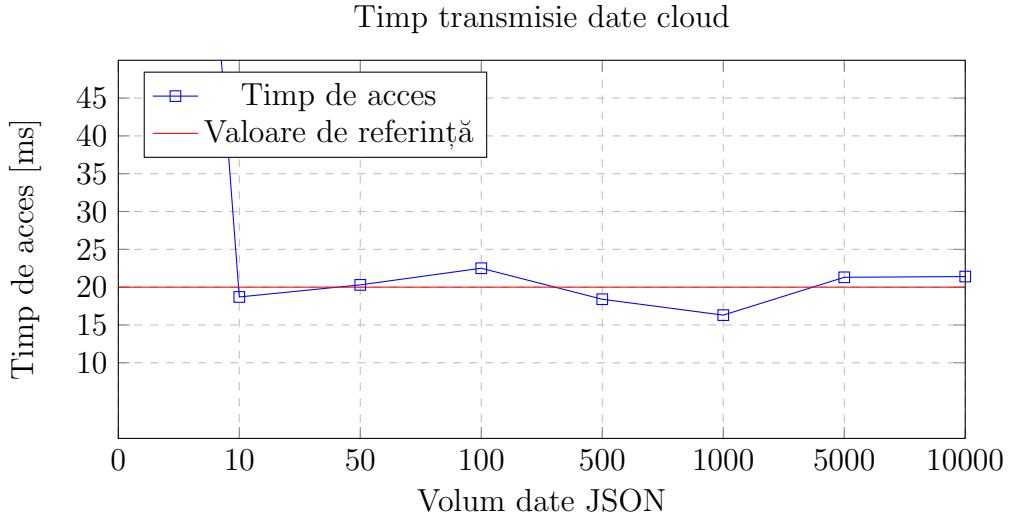


Figura 6.2: Stocarea datelor în contextul consolei AWS

O altă componentă în ceea ce privește performanța o constituie timpii calculați pentru achiziția datelor din Cloud și reprezentarea augmentată. Deși timpii introduși cu aplicația mobilă sunt considerabil mai mari în această fază, performanțele sunt mai mult decât acceptabile pentru un prototip în prima fază.

Așadar, valorile prezентate în Figura 6.3 sunt strict informative și ele pot varia la diferite utilizări ale aplicației.

6.1.2 Scalabilitate

Scalabilitatea reprezintă o specificație extrem de importantă în evaluarea unei soluții software. În ceea ce privește DynamoDB, Amazon insistă pe scalarea bazelor de date pe orizontală. Aceasta presupune scalarea bazelor de date pe mai multe noduri. DynamoDB suportă replicarea bazelor de date și partitioanarea acestora pentru viteze mari de scriere și citire. Amazon folosește la componenta fizică SSD (Solid-State-Drives) pentru hardware. Sistemul în sine este scalabil întrucât utilizând DynamoDB, un număr imens de microcontroleră poate fi atașat. În acest context, un număr și mai mare de familii și tipuri de senzori poate fi atașat acestor micro-controleră. Deși există limitări hardware, nivelul de scalabilitate oferit de Amazon pentru un astfel de sistem, este practic infinit.

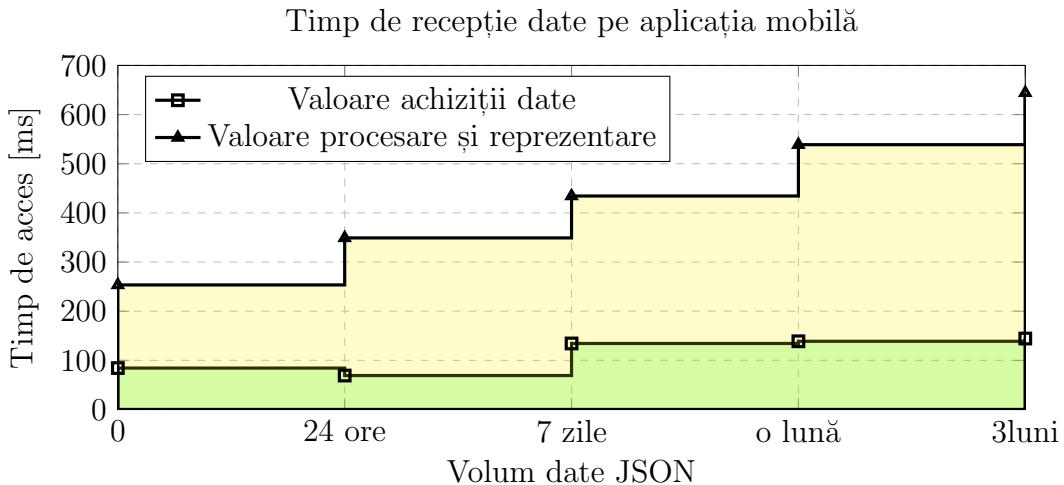


Figura 6.3: Achiziția datelor în contextul aplicației mobile

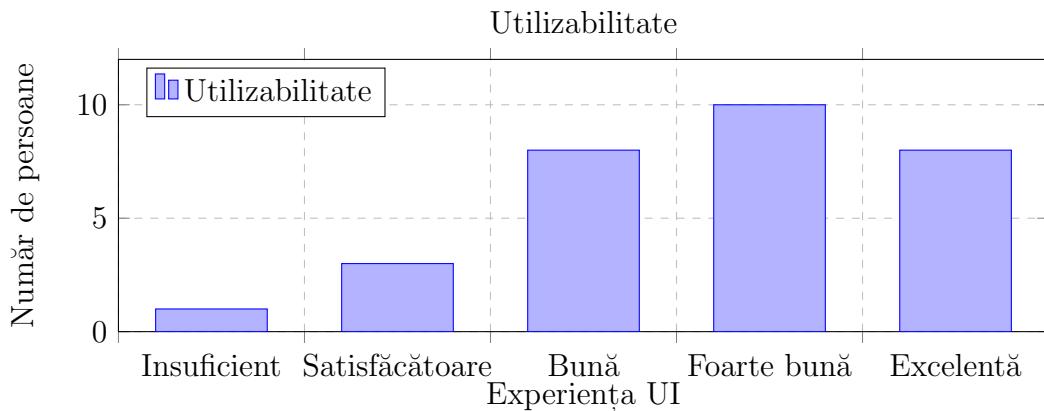


Figura 6.4: Utilizabilitate

6.1.3 Utilizabilitate

Utilizabilitatea este un atribut ce ține de calitatea software-ului și care determină cât de ușor este pentru utilizator să interacționeze cu interfața-utilizator. Este definită de componente de: capacitatea de învățare, eficiență, cât de ușor utilizatorul poate memora pașii interacțiunilor, erori și satisfacția folosind aplicația. În acest sens a fost efectuat un studiu asupra unui eșantion de 30 de persoane iar rezultatele au fost următoarele. Din 30 de persoane, 26 au avut o experiență bună, foarte bună sau excelentă (Figura 6.4). Deși aplicația este într-un stadiu incipient, rezultatele au fost relativ bune dar eșantionul este mic. Pe viitor, se sugerează identificarea unor caracteristici care ar îmbunătăți UI/UX și efectuarea unor studii detaliate pe eșantioane mai mari. Ceea ce rezultă din asta este faptul că aplicația are un nivel de calitate relativ bun.

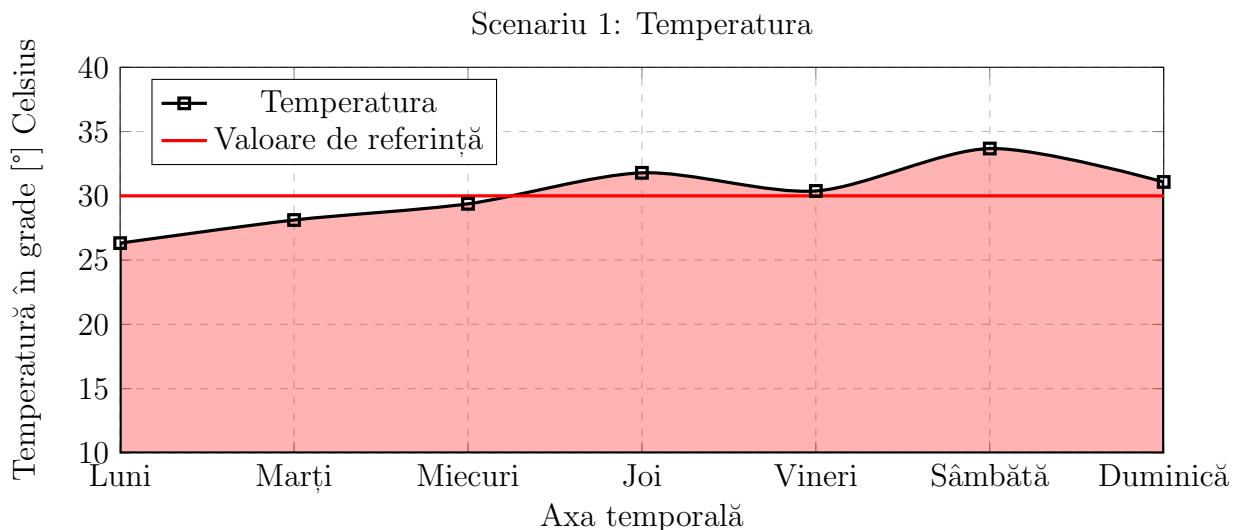


Figura 6.5: Temperatura pe durata unei săptămâni

6.1.4 Portabilitate

Portabilitatea este o formă specială de reutilizare. Unele componente software sunt mai puțin portabile decât altele. O aplicație software este considerată a fi portabilă pe un mediu nou dacă efortul necesar adaptării la un mediu este în limite rezonabile. În termeni de portabilitate, întrucât sistemul este bazat pe motorul de joc Unity, acesta poate fi implementat pe un număr de 27 de platforme distințe.

Pentru a dovedi aplicabilitatea arhitecturii în situații practice din lumea reală vor fi prezentate 3 scenarii în cele ce urmează.

6.2 Scenariu 1 - Casă automatizată

Primul scenariu constă în aplicarea arhitecturii într-un sistem de home-automation unde temperatura umiditatea, presiunea, nivelul de gaz, nivelul de alcool, luminozitate, etc. este monitorizat printr-o serie de 2 sau 3 microcontrollere Raspberry Pi cărora le sunt atașate aceste familii de senzori. Locatarii ajung acasă, scanează imaginea țintă și vizualizează graficele datelor senzoriale observând situații critice.

Figura 6.5 datele temporale într-un interval anume.

6.3 Scenariu 2 - Seră automatizată

Al doilea scenariu îl poate reprezenta monitorizarea datelor climatice dintr-o seră automatizată. Controale adiționale pentru menținerea condițiilor de seră pot fi adăugate și integrate extrem de ușor. Acestea ar putea fi ventilatoare, servo-motoare care să acționeze

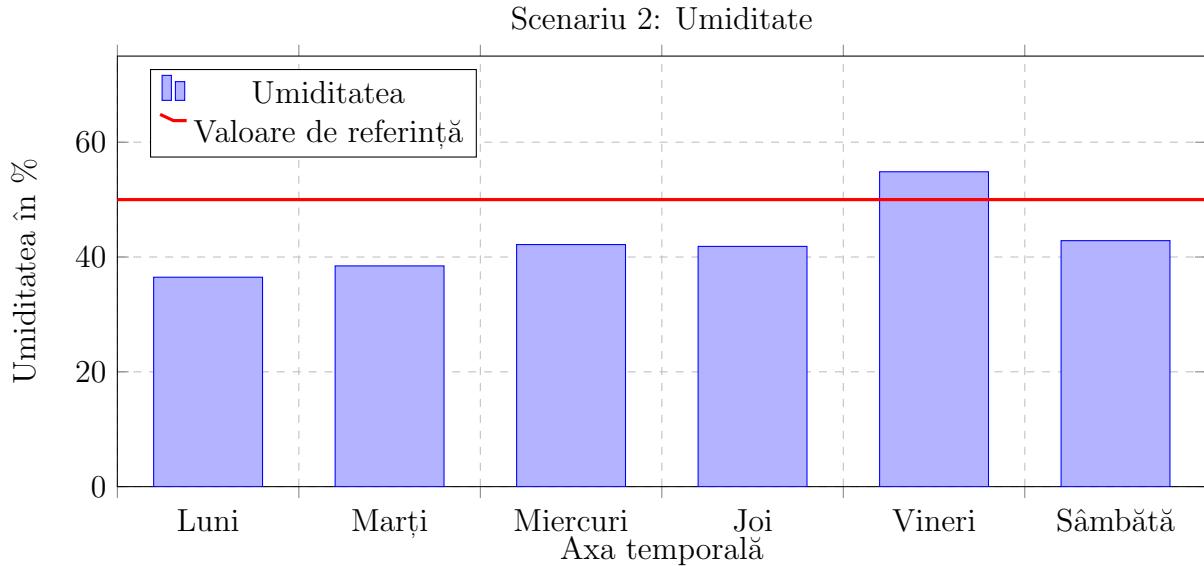


Figura 6.6: Umiditatea pe durata unei săptămâni

aceste ventilatoare, geamuri speciale care să se deschidă auto-comandat, nivelul de umezeală al solului, etc. Datele vor fi reprezentate prin aplicația mobilă. Desigur că îmbunătățiri pot fi aduse aplicației în aşa fel încât condițiile să poată fi acționate de pe telefonul mobil. Figura 6.6 reprezintă valori pe un interval dat.

6.4 Scenariu 3 - Centru de date

Al treilea scenariu ar putea fi un centru de date unde condițiile de mediu, temperatură, umiditate, presiune, volum de aer, etc. trebuie ținute sub strictă supraveghere. Cu arhitectura prezentată se elimină nevoia de a fi la fața locului, datele putând fi vizualizate și gestionate de la depărtare. De asemenea, pot fi introduse anumite acțiuni pentru a regla aceste condiții de mediu.

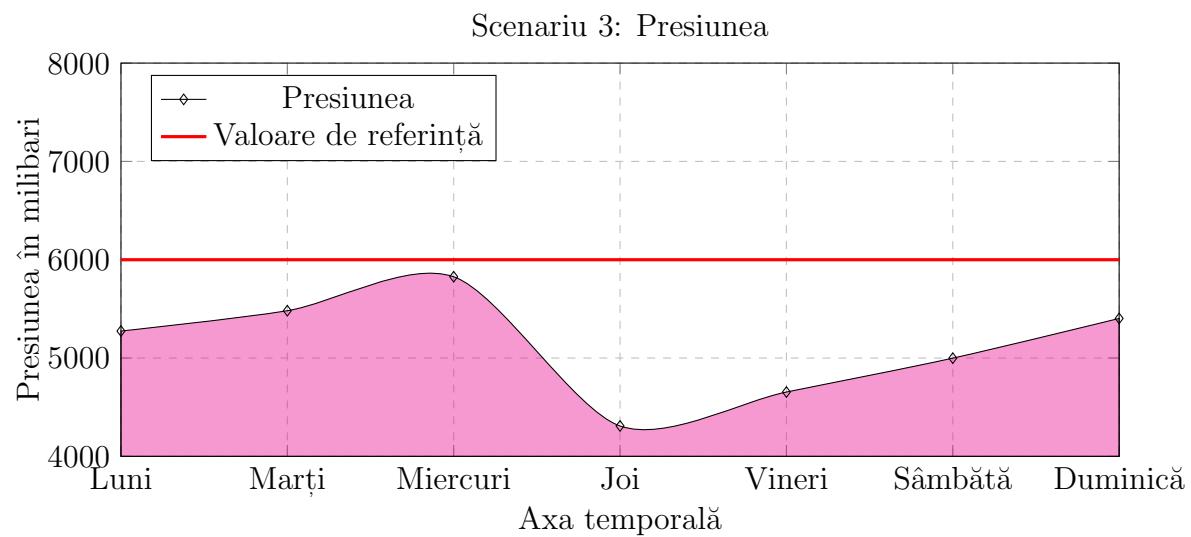


Figura 6.7: Presiunea pe durata unei săptămâni

Capitolul 7

Configurarea prototipului și manualul de utilizare

7.1 Instrucțiuni de instalare și configurare

Urmatoarele specificații trebuie configurate și îndeplinite pentru a reconstrui și implementa soluția:

1. Două microcontrollere Raspberry Pi
 - (a) Specificații Raspberry Pi 2 model B+
 - i. CPU: Broadcom BCM2837 ARM7 Quad Core Processor powered Single Board Computer at 900Mhz;
 - ii. GPU: Broadcom VideoCore IV 250 MHz; OpenGL ES 2.0 (24 GFLOPS); 1080p30 MPEG-2 și decodor VC-1; 1080p30 h.264/MPEG-4 AVC ;
 - iii. 40 pini extinși GPIO;
 - iv. Memorie: 1GB (comun cu GPU);
 - v. 4 porturi x USB 2;
 - vi. Full size HDMI;
 - vii. Input Video: 15-pin MIPI interfață cameră;
 - viii. Video output: HDMI, composite video (PAL and NTSC) via 3.5 mm jack;
 - ix. camera port CSI pentru conectarea Rasperi Pi-ului la cameră;
 - x. Audio input: I2S;
 - xi. Audio outputs: Analog via 3.5 mm jack; digital via HDMI and I2S;
 - xii. Rețea: 10/100Mbps Ethernet;
 - xiii. Stocare: Micro SD port pentru încărcarea sistemului de operare și pentru stocare;
 - xiv. Sursă alimentară: Micro USB 5 V and rating 800 mA (4.0 W);

7. Configurarea prototipului și manualul de utilizare

- (b) Specificații Raspberry Pi 3 model B
- i. CPU: Broadcom BCM2387 chipset. 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE);
 - ii. GPU: Dual Core VideoCore IV® Multimedia Co-Processor. Cu Open GL ES 2.0, accelerând hardware-ul OpenVG, și 1080p30 H.264. Capabil de 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs filtrare a texturilor și DMA
 - iii. Memorie: 1GB LPDDR2
 - iv. Connector GPIO : 40-pini 2.54 mm (100 mil) expansion header: 2x20 strip Cu 27 GPIO pini și +3.3 V, +5 V and GND supply lines
 - v. Video Output: HDMI (rev 1.3 and 1.4 Composite RCA (PAL and NTSC)
 - vi. Audio Output: Audio Output 3.5mm jack, HDMI USB 4 x USB 2.0 Connector
 - vii. 4 x USB 2 ports;
 - viii. Conector Cameră: 15-pin MIPI Interfață Serială Cameră (CSI-2)
 - ix. Conector Display: Display Serial Interface (DSI) 15 way
 - x. Stocare: Micro SD
 - xi. Rețea: 10/100Mbps BaseT Ethernet Socket;
 - xii. Sursă: Micro USB socket 5V1, 2.5A

(c)

2. Dispozitiv mobil Android

- (a) Motorola Moto X cu următoarele specificații:
- i. Sistem de Operare: Android 5.1 "Lollipop";
 - ii. CPU: 1.7 GHz dual-core Qualcomm Snapdragon S4 ProMobility; Motorola X8 Mobile Computing System (8 cores);
 - iii. GPU: Quad-core Adreno 320 @ 400 MHz;
 - iv. Memorie: 2 GB RAM;
 - v. Stocare: 16 GB;
 - vi. Sursă alimentare: 2200 mA;
 - vii. Camera Spate: 10 megapixels, 1080 HD video recording;
 - viii. Camera față: 2 megapixeli;

3. Router Linksys AC1750 Dual-Band Gigabit Cloud Wireless

- (a) Interfețe:
- i. 802.11 a/b/g/n/ac wireless LAN
 - ii. Four 10/100/1000 Gigabit LAN ports
 - iii. 10/100/1000 Gigabit WAN port

- iv. USB port
 - (b) IEEE 802.11ac (draft), IEEE 802.11n, IEEE 802.11g, IEEE 802.11b, IEEE 802.11a, IEEE 802.3, IEEE 802.3u
 - (c) Wireless Security
 - (d) WPA & WPA2 (Wi-Fi Protected Access)
 - (e) Wi-Fi Protected Setup (WPS) PIN/PBC

Următoarele software-uri trebuie instalate și configurate:

- Sistem de Operare: Raspbian Jessie – PIXEL (Debian Linux- based)
 - Configurarea sistemului de operare pe un card micro-SD boot-table;
- Activarea interfeței GPIO;
- Configurarea rețelei WiFi, determinarea IP-ului pentru Raspberry;
- Configurarea serverelor FTP, remote (Putty) și desktop visualisation (VNC);
- Python 2 și 3;
- Instalarea și configurarea librăriei SenseHAT pentru Python
- Configurarea bazei de date SQLite 3, deja existentă pe Raspberry Pi
- NPM (Node package manager) versiunea 3.10.8
- NodeJS versiunea 4.6.0
- Express JS
- SQLiteJS
- AWS Dynamo DB pentru NodeJs (JavaScript)

Pentru stocarea în serviciul de baze de date în cloud, există 2 soluții disponibile:

1. Testarea soluției într-un “cloud local” (instanța-serviciu care rulează local). Acest lucru este realizat prin intermediul unei interfețe web scrise în JavaScript care simulează serviciul de stocare NoSQL (recomandat). În consecință, librăria trebuie configurată pe o stație locală Windows care rulează sisteme de operare Windows 10/8/7.

7. Configurarea prototipului și manualul de utilizare

2. Crearea unui cont gratis AWS (Amazon Web Services) Free Tier. În acest context, utilizatorul/dezvoltatorul trebuie să țină cont să nu depăsească limita plafon impusă de Amazon relativ la operațiunile de citire/scriere în serviciu. Dacă această limită este depășită fiecare scriere/citire are un anumit cost monetar. În acest context este necesară autentificarea și autorizarea folosind polițele Cognito și IAM de configurare, pentru a oferi dispozitivului Raspberry PI și dispozitivului mobil acces pentru a achiziționa date din servicuiul de baze de date.

Aplicația de realitate augmentată a fost lansată pe un dispozitiv Android. Următoarele tehnologii și unelte trebuie instalate pe stația locală:

- Pachetul/extensia Vuforia pentru Unity (în interiorul proiectului Unity)
- Android Studio pentru singurul scop al instalării android SDK și dependințele aferente; Este nevoie de o atenție crescută la modificările ulterioare aduse sdk-ului și este posibil să existe anumite inconistențe referitoare la namespace și la dependințe.
- AWS DynamoDB pentru Unity/.NET tools;
- Pachetul Unity Pro Chart pentru reprezentarea și vizualizarea datelor;
- Activarea modului "Dezvoltator" pentru telefonul mobil.

Script-urile Python pentru date senzoriale și cele pentru serverul de transmisie în cloud trebuie configurate și rulate pe Raspberry Pi. De asemenea, script-uri adiționale sunt folosite pentru a crea baza de date locală SQLite. Librăriile Node dependente sunt gestionate și rezolvate de npm la initializare și crearea pachetului. Instanța locală de AWS DynamoDB poate crea tabelul de baze de date direct din interfața web folosind sintaxa Dynamo. Proiectul Unity conține toate fișierele necesare pentru configurare și rulare.

Cerințe preliminare:

- Cunoștințe de bază cu privire la micro-controllere (hardware, electronică și circuite electrice);
- Cunoștințe de bază cu privire la distribuțiile Linux (preferabil Debian/Raspbian);
- Cunoștințe de bază în Python;
- Cunoștințe de bază și intermediare în JavaScript și NodeJs;
- Cunoștințe și experiență bazică în servicii web;
- Cunoștințe intermediare și avansate de Unity 3D;
- Cunoștințe intermediare și avansate C#;
- Cunoștințe de bază cu Vuforia AR;
- Cunoștințe de bază în Android;

7.2 Instructiuni de utilizare

Odată încărcată aplicația pe mobil, aplicația poate fi utilizată în următorul mode. Diagrama 5.2 cuprinde următoarele acțiuni exemplificate prin capturi de imagini:

În cele ce urmează, se va descrie pas cu pas setul de acțiuni posibile în momentul de față pe aplicația mobilă Android din perspectiva utilizatorului final. În momentul lansării aplicației, utilizatorul deschide aplicația:

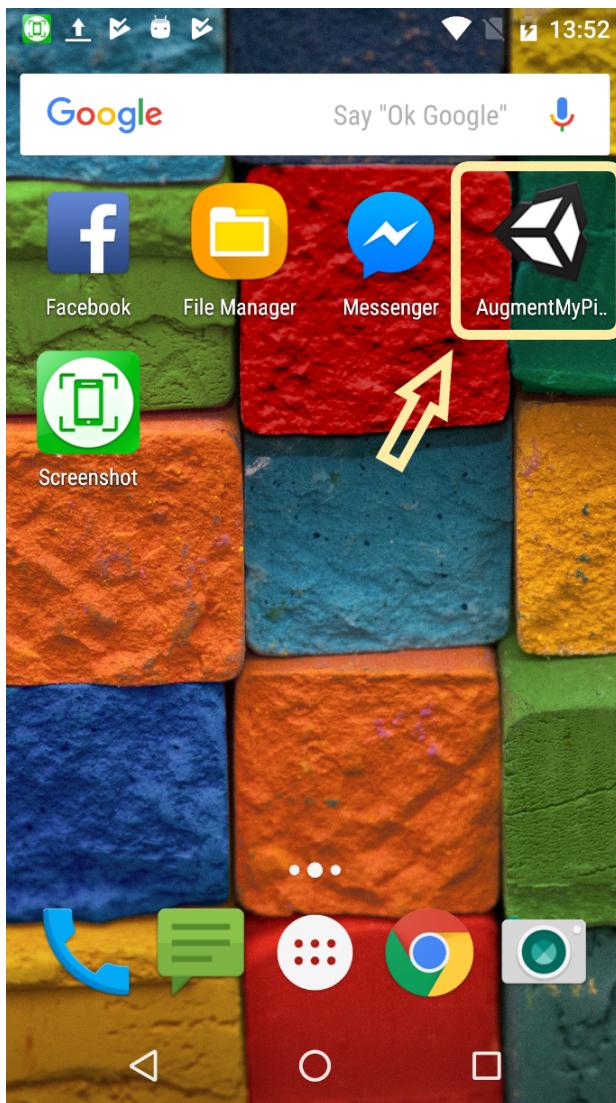


Figura 7.1: Aplicația pe Android Motorola Moto X

Apoi, odată lansată aplicația, imaginea țintă trebuie scanată:

După ce ținta este scanată, două butoane vor apărea încadrate în obiectul 3-D.

Următorul pas constă în achiziționarea datelor prin butonul "Get Data" iar apoi

7. Configurarea prototipului și manualul de utilizare

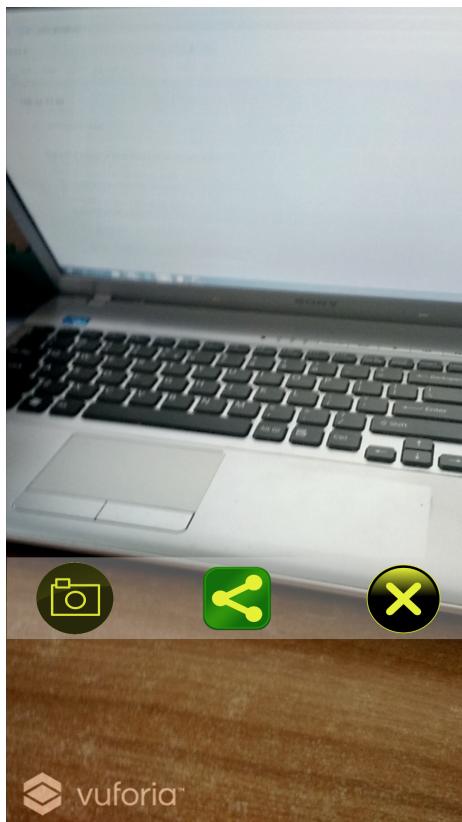


Figura 7.2: Aplicația înainte de scanarea țintei

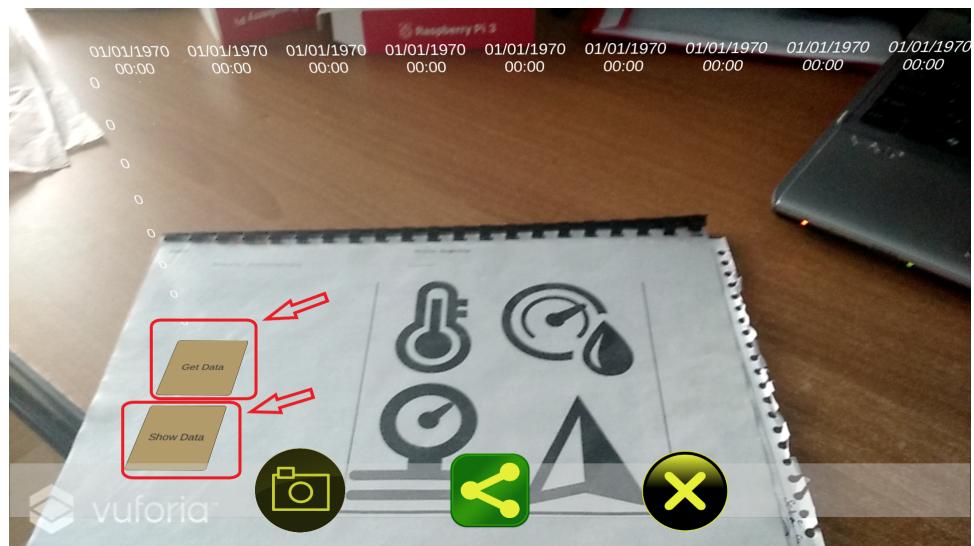


Figura 7.3: Aplicația după scanarea țintei

7.2. Instructiuni de utilizare

reprezentarea prin butonul "View Data".



Figura 7.4: Obiectul cu datele 3D mapate pe țintă

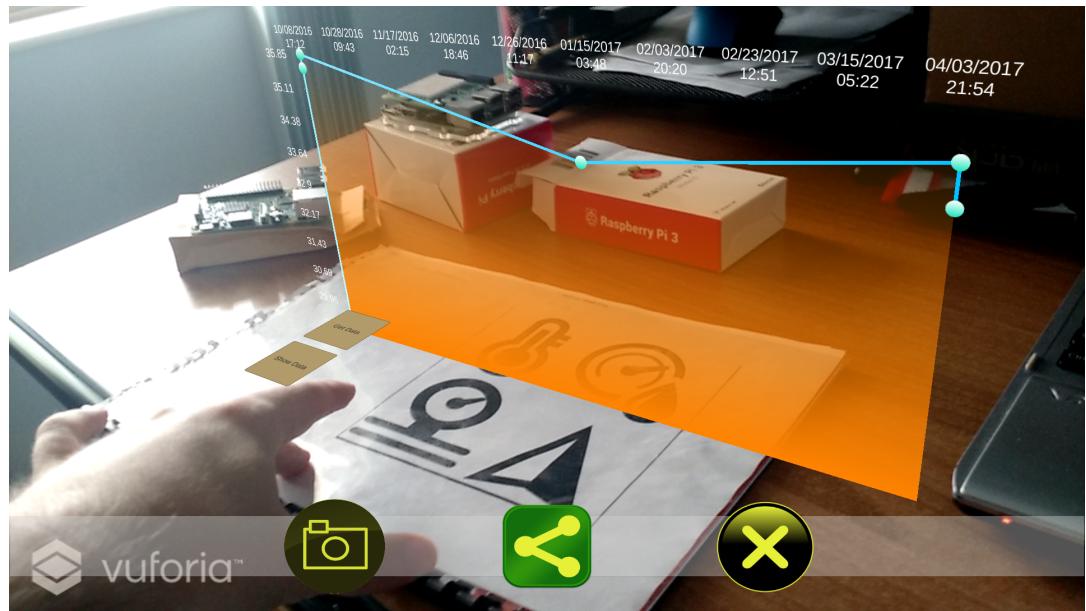


Figura 7.5: Obiectul rotit cu datele 3D mapate pe țintă

Apoi, dacă utilizatorul dorește, poate face o captură foto a ecranului cu graficul. În final, aplicația e închisă prin butonul de închidere.

7. Configurarea prototipului și manualul de utilizare



Figura 7.6: Reprezentare în modul portret

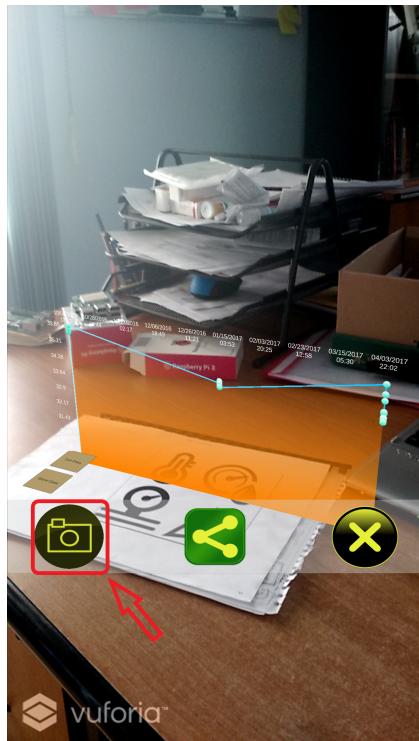


Figura 7.7: Captura ecranului cu graficul virtual



Figura 7.8: Închiderea aplicației

Capitolul 8

Concluzii

8.1 Rezumatul realizărilor

Teza prezintă reprezentă un proiect de cercetare, inginerie și explorare a cărui obiectiv principal este elaborarea, studierea și dezvoltarea unei soluții colaborative și sinergetice unde datele heterogene IoT sunt achiziționate și procesate prin intermediul unu microcontroller. Datele sunt apoi transmise printr-un serviciu REST spre serviciul de baze de date din cloud unde sunt persistate și salvate și apoi preluate de aplicația mobilă unde sunt reprezentate sub forma unui obiect 3D augmentat.

Unul dintre factorii determinanți și cruciali care stau la baza întregului proces de cercetare, modelare, proiectare, implementare și testare a fost elaborarea unei soluții prototip care să înglobeze domeniile din știința calculatoarelor menționate în capitolul 4.

Studiul teoretic și bibliografic a constat în achiziționarea, filtrarea și sintetizarea literaturii de specialitate și actualitate(cărți, articole, documente tehnice, expertize, exemple online, video-uri și prezentări). Bineînțeles că acomodarea cu vocabularul, conceptele, analizele informației, selecția și extracția au rezultat ca urmare a unu studiu amănunțit.

Din punctul de vedere al implementării, a fost o provocare realizarea coordonării și cooperării tehnologiilor enunțate la capitolul 5. Această soluție oferă un început în această direcție a AR-Cloud-IoT. Cu toate acestea, trebuie considerată ca și "Work-In-Progress" devreme ce îmbunătățirile care pot fi aduse sunt extrem de variate.

În consecință, obiectivul principal a fost îndeplinit. De asemenea, realizările implică și studiul tehnologiilor: SQLite, Python, NodeJs, AWS DynamoDb, Unity, Vuforia și Android.

Soluția a fost validată și evaluată prin scenarii experimentale prezentate în detaliu în capitolul 6, fiind bazate pe niște metriki predefinite. Testele au dus la concluzia că soluția este nu doar fezabilă, scalabilă, utilizabilă dar și eficientă și portabilă.

După cum dovedește și studiul prezent, soluțiile care vizează această sinergie, cooperare și inter-operabilitate între cele 3 nivele, sunt aproape inexistente. Nicunul nu se axează pe experiența utilizatorului, pe utilizabilitate sau pe interacțiune. Din aceste motive, partea inovativă constituie această încadrare a componentei de realitate augmen-

8. Concluzii

tată cu Cloud și IoT, reprezentând de asemenea și un fundament important pentru studii viitoare. Nu în ultimul rând, decuplarea componentelor face extrem de ușoară injectarea altor tehnologii curente sau viitoare în arhitectura experimentală.

Directii cercetare	State of the art	Contributia
Integrarea Realitatii Augmentata cu Cloud IoT	Abordările recente oferă integrarea IoT cu infrastructurile Cloud cât și Cloud cu Realitate Augmentată	Dezvoltarea unei soluții care integrează achiziția de date de pe dispozitive IoT, o stochează în Cloud reprezintă datele în contextul Realitatii augmentate

Tabela 8.1: Contribuții personale

8.2 Îmbunătățiri viitoare

1. Testare mai detaliată cu scenarii complexe din lumea reală;
2. Atașarea mai multor tipuri de dispozitive (Arduino, LattePanda sau Tessel);
3. Explorarea diferitor tipuri de senzori față de cei deja folosiți: biometric, RFID, forță, infraroșu, magnetic, sonar, video, audio, etc. și prezentarea unor scenarii plauzibile;
4. Integrarea altor baze de date NoSQL din cadrul AWS în proiect;
5. Explorarea posibilității de integrare a serviciului AWS IoT pentru dispozitive conectate;
6. Explorarea serviciului AWS Identity and Access management (IAM) pentru controlul accesului securizat la serviciile AWS;
7. Explorarea și studiul în continuare a librăriei Vuforia AR și integrarea ei cu AWS;
8. Integrarea și dezvoltarea sau modificarea librăriilor pentru reprezentare și vizualizare 3D;
9. Realizarea unei interfețe UI pe sistemul de operare Android care să vizeze experiența utilizatorului din considerente de simplicitate și flexibilitate;
10. Noi controale UI și alte caracteristici;
11. Cercetarea posibilității de elaborare a unui scenariu aplicativ de business;

Bibliografie

- [1] K. Rose, S. Eldridge, and L. Chapin, “The internet of things: An overview - understanding the issues and challenges of a more connected world,” 2015.
- [2] D. Evans, “The internet of things - how the next evolution of the internet is changing everything,” April 2011.
- [3] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: state-of-the-art and research challenges,” *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, May 2010. [Online]. Available: <https://doi.org/10.1007/s13174-010-0007-6>
- [4] D. Corporation, “Introduction to cloud computing.”
- [5] R. T. Azuma, “A survey of augmented reality,” *Presence: Teleoper. Virtual Environments*, vol. 6, no. 4, pp. 355–385, Aug. 1997. [Online]. Available: <http://dx.doi.org/10.1162/pres.1997.6.4.355>
- [6] S. Siltanen and V. teknillinen tutkimuskeskus, *Theory and Applications of Marker-based Augmented Reality*, ser. VTT science. JULKAISIJA – UTGIVARE, 2012. [Online]. Available: <https://books.google.ro/books?id=dbtUlQEACAAJ>
- [7] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo, “A survey on facilities for experimental internet of things research,” *IEEE Communications Magazine*, vol. 49, no. 11, pp. 58–67, November 2011.
- [8] “Understanding the internet of things (iot),” July 2014.
- [9] H.-D. Ma, “Internet of things: Objectives and scientific challenges,” *Journal of Computer Science and Technology*, vol. 26, no. 6, pp. 919–924, Nov 2011. [Online]. Available: <https://doi.org/10.1007/s11390-011-1189-5>
- [10] A. Whitmore, A. Agarwal, and L. Da Xu, “The internet of things—a survey of topics and trends,” *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, Apr 2015. [Online]. Available: <https://doi.org/10.1007/s10796-014-9489-2>
- [11] S. Hong, D. Kim, M. Ha, S. Bae, S. J. Park, W. Jung, and J. e. Kim, “Snail: an ip-based wireless sensor network approach to the internet of things,” *IEEE Wireless Communications*, vol. 17, no. 6, pp. 34–42, December 2010.

Bibliografie

- [12] D. Miorandi, S. Sicari, F. D. Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497 – 1516, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870512000674>
- [13] L. Atzori, A. Iera, and G. Morabito, "From "smart objects" to "social objects": The next evolutionary step of the internet of things," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 97–105, January 2014.
- [14] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *IEEE Transactions on Services Computing*, vol. 3, pp. 223–235, 2010.
- [15] D. Le-Phuoc, H. N. M. Quoc, H. N. Quoc, T. T. Nhat, and M. Hauswirth, "The graph of things: A step towards the live knowledge graph of connected things," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 37, pp. 25 – 35, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570826816000196>
- [16] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "On the integration of cloud computing and internet of things," in *Proceedings of the 2014 International Conference on Future Internet of Things and Cloud*, ser. FICLOUD '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 23–30. [Online]. Available: <http://dx.doi.org/10.1109/FiCloud.2014.14>
- [17] "Internet of things and cloud: Converging the opposites towards the future internet," May 2012.
- [18] D. Huang, T. Xing, and H. Wu, "Mobile cloud computing service models: a user-centric approach," *IEEE Network*, vol. 27, no. 5, pp. 6–11, September 2013.
- [19] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb 2009. [Online]. Available: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>
- [20] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1721654.1721672>
- [21] Y. Liu, B. Dong, B. Guo, J. Yang, and W. Peng, "Combination of cloud computing and internet of things (iot) in medical monitoring systems," *International Journal of Hybrid Information Technology*, vol. 8, pp. 367–376, 12 2015.

- [22] R. Buyya, C. S. Yeo, and S. Venugopal, “Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities,” in *Proceedings of the 2008 10th IEEE International Conference on High Performance Computing and Communications*, ser. HPCC ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 5–13. [Online]. Available: <http://dx.doi.org/10.1109/HPCC.2008.172>
- [23] S. M. B. Amol S. Choure, “Cloud computing: state-of-the-art and research challenges,” *International Journal of Computer Applications*, vol. 122, no. 18, 2015.
- [24] J. Wu, L. Ping, X. Ge, Y. Wang, and J. Fu, “Cloud storage as the infrastructure of cloud computing,” in *2010 International Conference on Intelligent Computing and Cognitive Informatics*, June 2010, pp. 380–383.
- [25] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z. L. Zhang, “Unreeling netflix: Understanding and improving multi-cdn movie delivery,” in *2012 Proceedings IEEE INFOCOM*, March 2012, pp. 1620–1628.
- [26] A. Nayak, A. Poriya, and D. Poojary, “Article: Type of nosql databases and its comparison with relational databases,” *International Journal of Applied Information Systems*, vol. 5, no. 4, pp. 16–19, March 2013, published by Foundation of Computer Science, New York, USA.
- [27] K. Grolinger, W. A. Higashino, A. Tiwari, and M. A. Capretz, “Data management in cloud environments: Nosql and newsql data stores,” *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 2, no. 1, p. 22, December 2013. [Online]. Available: <https://doi.org/10.1186/2192-113X-2-22>
- [28] J. Kepner, V. Gadepally, D. Hutchison, H. Jananthan, T. G. Mattson, S. Samsi, and A. Reuther, “Associative array model of sql, nosql, and newsql databases,” *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–9, 2016.
- [29] N. R. Council, *Virtual Reality: Scientific and Technological Challenges*, N. I. Durlach and A. S. Mavor, Eds. Washington, DC, USA: National Academy Press, 1994.
- [30] W. E. L. Grimson, T. Lozano-Perez, S. J. White, W. M. I. Wells, R. Kikinis, and G. J. Ettinger, “An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization,” in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Jun 1994, pp. 430–436.
- [31] W. Lorensen, H. Cline, C. Nafis, R. Kikinis, D. Altobelli, and L. Gleason, “Enhancing reality in the operating room,” in *Visualization, 1993. Visualization ’93, Proceedings., IEEE Conference on*, Oct 1993, pp. 410–415.
- [32] F. Betting, J. Feldmar, N. Ayache, and F. Devernay, *A New Framework for Fusing Stereo Images with Volumetric Medical Images*. Berlin, Heidelberg:

- Springer Berlin Heidelberg, 1995, pp. 30–39. [Online]. Available: https://doi.org/10.1007/978-3-540-49197-2_4
- [33] S. Feiner, B. Macintyre, and D. Seligmann, “Knowledge-based augmented reality,” *Commun. ACM*, vol. 36, no. 7, pp. 53–62, Jul. 1993. [Online]. Available: <http://doi.acm.org/10.1145/159544.159587>
- [34] J. Rekimoto and K. Nagao, “The world through the computer: Computer augmented interaction with real world environments,” in *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology*, ser. UIST ’95. New York, NY, USA: ACM, 1995, pp. 29–36. [Online]. Available: <http://doi.acm.org/10.1145/215585.215639>
- [35] P. Milgram, S. Zhai, D. Drascic, and J. Grodski, “Applications of augmented reality for human-robot communication,” in *Intelligent Robots and Systems ’93, IROS ’93. Proceedings of the 1993 IEEE/RSJ International Conference on*, vol. 3, Jul 1993, pp. 1467–1472 vol.3.
- [36] *Virtual window telepresence system for telerobotic inspection*, vol. 2351, 1995. [Online]. Available: <http://dx.doi.org/10.1117/12.197330>
- [37] H.-K. Wu, S. W.-Y. Lee, H.-Y. Chang, and J.-C. Liang, “Current status, opportunities and challenges of augmented reality in education,” *Comput. Educ.*, vol. 62, pp. 41–49, Mar. 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.compedu.2012.10.024>
- [38] M. Dunleavy and R. Dede, Chrisand Mitchell, “Affordances and limitations of immersive participatory augmented reality simulations for teaching and learning,” *Journal of Science Education and Technology*, vol. 18, no. 1, pp. 7–22, Feb 2009. [Online]. Available: <https://doi.org/10.1007/s10956-008-9119-1>
- [39] M. E. C. Santos, A. Chen, T. Taketomi, G. Yamamoto, J. Miyazaki, and H. Kato, “Augmented reality learning experiences: Survey of prototype design and evaluation,” *IEEE Transactions on Learning Technologies*, vol. 7, no. 1, pp. 38–56, Jan 2014.
- [40] K.-H. Cheng and C.-C. Tsai, “Affordances of augmented reality in science learning: Suggestions for future research,” *Journal of Science Education and Technology*, vol. 22, no. 4, pp. 449–462, Aug 2013. [Online]. Available: <https://doi.org/10.1007/s10956-012-9405-9>
- [41] M. B. Ibáñez, A. J. d. Castro, and C. D. Kloos, “An empirical study of the use of an augmented reality simulator in a face-to-face physics course,” in *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*, July 2017, pp. 469–471.

- [42] G. W. Ng, Y. Oon, H. Y. Lee, and E. H. Teoh, “An augmented reality system for biology science education in malaysian.” *International Journal of Innovative Computing*, vol. 6, no. 2, pp. 8–13, 2013. [Online]. Available: <http://ir.unimas.my/id/eprint/15857>
- [43] T. H.-C. Chiang, S. J. H. Yang, and G.-J. Hwang, “An augmented reality-based mobile learning system to improve students’ learning achievements and motivations in natural science inquiry activities,” *Educational Technology and Society*, vol. 17, pp. 352–365, 2014.
- [44] A. Di Serio, M. B. Ibáñez, and C. D. Kloos, “Impact of an augmented reality system on students’ motivation for a visual art course,” *Computers and Education*, vol. 68, pp. 586–596, Oct. 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.compedu.2012.03.002>
- [45] D. Fonseca, N. Martí, E. Redondo, I. Navarro, and A. Sánchez, “Relationship between student profile, tool use, participation, and academic performance with the use of augmented reality technology for visualized architecture models,” *Comput. Hum. Behav.*, vol. 31, pp. 434–445, Feb. 2014. [Online]. Available: <http://dx.doi.org/10.1016/j.chb.2013.03.006>
- [46] K. Chang, C. Chang, H. Hou, Y. Sung, H. Chao, and C. Lee, “Development and behavioral pattern analysis of a mobile guide system with augmented reality for painting appreciation instruction in an art museum,” *Computers and Education*, vol. 71, pp. 185–197, 2014.
- [47] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway, “Exploring mars: Developing indoor and outdoor user interfaces to a mobile augmented reality system,” *Computers and Graphics*, vol. 23, pp. 779–785, 1999.
- [48] F. Steven and M. B. H. T. W. Anthony, “A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment,” *Personal Technologies*, vol. 1, no. 4, pp. 208–217, Dec 1997. [Online]. Available: <https://doi.org/10.1007/BF01682023>
- [49] P. Geiger, M. Schickler, R. Pryss, J. Schobel, and M. Reichert, “Location-based mobile augmented reality applications: Challenges, examples, lessons learned,” in *10th Int'l Conference on Web Information Systems and Technologies (WEBIST 2014), Special Session on Business Apps*, April 2014, pp. 383–394. [Online]. Available: <http://dbis.eprints.uni-ulm.de/1028/>
- [50] D. Amin and S. Govilkar, “Comparative study of augmented reality sdk’s,” in *International Journal on Computational Science & Applications*, 2015.
-

Bibliografie

- [51] F. Peng and J. Zhai, “A mobile augmented reality system for exhibition hall based on vuforia,” in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, June 2017, pp. 1049–1052.
- [52] C. Xiao and Z. Lifeng, “Implementation of mobile augmented reality based on vuforia and rawajali,” in *2014 IEEE 5th International Conference on Software Engineering and Service Science*, June 2014, pp. 912–915.
- [53] H. Tschofenig, J. Arkko, D. Thaler, and D. R. McPherson, “Architectural Considerations in Smart Object Networking,” RFC 7452, Mar. 2015.
- [54] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, “Recent advances in augmented reality,” *IEEE Comput. Graph. Appl.*, vol. 21, no. 6, pp. 34–47, Nov. 2001. [Online]. Available: <http://dx.doi.org/10.1109/38.963459>
- [55] V. Andreies, “Game-oriented tools layered on unity 3d,” July 2013, lucrare de licență.

Anexa A

Fragmente cod sursă

A.1 Aplicația mobilă - C#

Listing A.1: SensorRepository.cs

```
1  public class SensorRepository : IRepository<SensorReading>
2  {
3      private static AmazonDynamoDBClient _client;
4      private AmazonDynamoDBConfig _config;
5      private DynamoDBContext _context;
6
7      private IList<SensorReading> sensorReadings;
8      private SensorReading _sensorLog;
9
10     public SensorRepository()
11     {
12         _config = new AmazonDynamoDBConfig();
13         _config.ServiceURL = "http://192.168.0.100:8000/";
14         _client = new AmazonDynamoDBClient("fakeMyKeyId", "fakeSecretAccessKey", _config);
15         _context = new DynamoDBContext(_client);
16     }
17     public void BatchStore(IEnumerable<SensorReading> items)
18     {
19         throw new NotImplementedException();
20     }
21
22     public void Delete(SensorReading item)
23     {
24         _context.DeleteAsync<SensorReading>(item.Id, (res)=> {
25             if (res.Exception == null)
26             {
27                 _context.LoadAsync<SensorReading>(item.Id, (result)=> {
28                     SensorReading deletedReading = result.Result;
```

A. Fragmente cod sursă

```
30             if (deletedReading == null) Debug.Log("Item was
31                     deleted");
32         });
33     });
34 }
35
36
37     public IList<SensorReading> GetAll(Dictionary<string , AttributeValue
38 > lastKeyEvaluated)
39     {
40         var request = new ScanRequest
41         {
42             TableName = "SensorReadings",
43             Limit = 100,
44             ExclusiveStartKey = lastKeyEvaluated ,
45             };
46
47         sensorReadings = new List<SensorReading>();
48
49         _client.ScanAsync(request , (result) =>
50         {
51             foreach (Dictionary<string , AttributeValue> item
52                     in result.Response.Items)
53             {
54                 _sensorLog = new SensorReading();
55
56                 ... parsing values
57             }
58             sensorReadings.Add(_sensorLog);
59         }
60         lastKeyEvaluated = result.Response.LastEvaluatedKey;
61         if (lastKeyEvaluated != null && lastKeyEvaluated.Count != 0)
62         {
63             Debug.Log("*** The last key evaluated is " +
64                         lastKeyEvaluated);
65             GetAll(lastKeyEvaluated);
66         }
67     });
68
69     return sensorReadings;
70 }
71
72     public SensorReading GetItem(string key)
73     {
74         SensorReading queriedItem = null;
75         _context.LoadAsync<SensorReading>(key , (result) =>
76         {
```

```

77         if (result.Exception == null)
78     {
79         queriedItem = result.Result as SensorReading;
80         Debug.Log("Item was retrieved");
81     }
82 });
83 return queriedItem;
84 }

85
86 public void Store(SensorReading item)
87 {
88     _context.SaveAsync(item, (result) => {
89         if (result.Exception == null)
90             Debug.Log("Saved Sensor Value");
91     });
92 }

93
94 public void Update(SensorReading item)
95 {
96     SensorReading updatedReading = null;
97     _context.LoadAsync<SensorReading>(item.Id, (result) =>
98     {
99         if (result.Exception == null)
100     {
101         updatedReading = result.Result as SensorReading;
102         // Update item
103         _context.SaveAsync<SensorReading>(updatedReading, (res)
104             =>
105             {
106                 if (res.Exception == null)
107                     Debug.Log("Updated Sensor Value");
108             });
109     });
110 }

111
112 #region IDisposable Support
113 private bool disposedValue = false; // To detect redundant calls
114
115 protected virtual void Dispose(bool disposing)
116 {
117     if (!disposedValue)
118     {
119         if (disposing)
120         {
121             // TODO: dispose managed state (managed objects).
122         }
123
124         // TODO: free unmanaged resources (unmanaged objects) and
125         // override a finalizer below.
126     }
127 }

```

A. Fragmente cod sursă

```
125         // TODO: set large fields to null.
126
127         disposedValue = true;
128     }
129 }
130
131 // TODO: override a finalizer only if Dispose(bool disposing) above
132 // has code to free unmanaged resources.
133 // ~SensorRepository() {
134 //     // Do not change this code. Put cleanup code in Dispose(bool
135 //     // disposing) above.
136 //     // Dispose(false);
137 // }
138
139 // This code added to correctly implement the disposable pattern.
140 public void Dispose()
141 {
142     // Do not change this code. Put cleanup code in Dispose(bool
143     // disposing) above.
144     Dispose(true);
145     // TODO: uncomment the following line if the finalizer is
146     // overridden above.
147     // GC.SuppressFinalize(this);
148 }
149 #endregion
150 }
```

Listing A.2: TemperatureChartFeedController.cs

```

1  public class TemperatureChartFeedController : MonoBehaviour
2  {
3      private SensorRepository _sensorRepository;
4      private GraphChartBase _graph;
5      public Button getSensorData;
6      public Button showData;
7      private IList<SensorReading> sensorReadings;
8
9
10     void Start()
11     {
12         Amazon.UnityInitializer.AttachToGameObject(this.gameObject);
13         _sensorRepository = new SensorRepository();
14         getSensorData.onClick.AddListener(PerformReadOnTableListener);
15         showData.onClick.AddListener(ConstructGraph);
16     }
17
18     private void ConstructGraph()
19     {
20         _graph = GetComponent<GraphChartBase>();
21
22         if (_graph != null && sensorReadings.Count != 0)
23         {
24             _graph.DataSource.StartBatch();
25             _graph.DataSource.ClearCategory("Temperature");
26             for (int i = 0; i < sensorReadings.Count - 1; i++)
27             {
28                 _graph.DataSource.AddPointToCategory("Temperature",
29                     Convert.ToDateTime(sensorReadings[i].Date),
30                     sensorReadings[i].Temperature);
31             }
32         }
33     }
34
35     private void PerformReadOnTableListener()
36     {
37         sensorReadings = _sensorRepository.GetAll(null);
38     }
}

```

A.2 Achiziție valori senzor - Python

```
1 #!/usr/bin/python
2 import os
3 import time
4 import glob
5 import Sensors_Repository as rep
6 from sense_hat import SenseHat
7
8
9 # global variables – intervals between readings
10 SECONDS_60=60
11 MINUTES_45=45 * SECONDS_60
12 MINUTES_60=60 * SECONDS_60
13
14 # initialize the sense HAT Add-On
15 sense = SenseHat()
16
17 # created the database and the underlying table
18 rep.create_table()
19
20 # read the temperature, humidity and pressure from the HAT
21 # prepare table line
22 # round-up the values and save them
23 while True:
24     table_row = []
25
26     # read each value
27     temperature = sense.get_temperature()
28     humidity = sense.get_humidity()
29     pressure = sense.get_pressure()
30
31     # construct the sensor list
32     table_row.append(round(temperature,2))
33     table_row.append(round(humidity,2))
34     table_row.append(round(pressure,2))
35
36     # store values in database
37     rep.data_insert(table_row[0], table_row[1], table_row[2])
38
39     # print the values in the terminal
40     print("_____")
41     print("Temperature value is = %s C" % temperature)
42     print("Humidity value is = %s %rH" % humidity)
43     print("Pressure value is = %s Millibars" % pressure)
44     print("_____")
45
46     # interval – wait for
47     time.sleep(SECONDS_60)
48
```

```

49     # print on the HAT display
50     sense.show_message("Temp %s" % table_row[0])
51     sense.show_message("Hum %s" % table_row[1])
52     sense.show_message("Press %s" % table_row[2])

```

Listing A.3: SensorLogger.py

```

1 import os
2 import sqlite3
3 from SensorNo import s_01
4 from SensorTypes import ENVIRONMENT_01
5
6
7 db_filename = '/home/pi/Documents/Personal Projects/Cloud Sensors Data
8 Logger/SQLITE_DB/Sensors.db'
9 db_is_new = not os.path.exists(db_filename)
10
11 def create_table():
12     try:
13         conn = sqlite3.connect(db_filename)
14         c = conn.cursor()
15         create_stmt = 'CREATE TABLE IF NOT EXISTS
16             SensorReadings( Id INTEGER PRIMARY KEY AUTOINCREMENT \
17                 NOT NULL \
18                 UNIQUE,
19                 Sensor_No INTEGER, \
20                 Sensor_Type CHAR(30), \
21                 Temperature DOUBLE, \
22                 Humidity DOUBLE, \
23                 Pressure DOUBLE, \
24                 GPS CHAR, \
25                 Datestamp DATETIME )'
26         c.execute(create_stmt)
27         if db_is_new:
28             print('Schema does not exist. Creating it ... ')
29         else:
30             print('Database already exists. It is assumed that schema also
exists.')
31         conn.commit()
32     except Exception as ex:
33         conn.rollback()
34         raise ex
35     finally:
36         c.close()
37         conn.close()
38
39 def data_insert(temp, hum, lum):
40     try:
41         conn = sqlite3.connect(db_filename)
42         c = conn.cursor()
43         insert_stmt = "INSERT INTO SensorReadings (Sensor_No, Sensor_Type ,

```

A. Fragmente cod sursă

```
43     Temperature , Humidity , Pressure , Datestamp) \          VALUES ((?) , (?) , (?) , (?) , (?) ,
44     DATETIME( 'now' , 'localtime' ))"                  conn.execute(insert_stmt , [s_01 , ENVIRONMENT_01, temp , hum , lum])
45     c . execute(insert_stmt , [s_01 , ENVIRONMENT_01, temp , hum , lum])
46     conn.commit()
47 except Exception as ex:
48     conn.rollback()
49     raise ex
50 finally:
51     c . close()
52     conn.close()
```

Listing A.4: SensorRepository.py

A.3 Transmisie valori senzori - JavaScript/NodeJs

```
1  /**
2   * Created by Virgil on 11.03.2017
3   */
4
5 // Require modules
6 var sqlite3 = require('sqlite3').verbose();
7 var AWS = require("aws-sdk");
8
9 //AWS Cloud Configuration
10 //run on local DynamoDB instance
11 AWS.config.update({
12     region: "eu-central-1",
13     endpoint: "http://192.168.0.100:8000/",
14     // accessKeyId default can be used while using the downloadable version
15     // For security reasons, do not store AWS Credentials in your files. Use
16     // Amazon Cognito instead.
17     accessKeyId: "fakeMyKeyId",
18     // secretAccessKey default can be used while using the downloadable
19     // For security reasons, do not store AWS Credentials in your files. Use
19     secretAccessKey: "fakeSecretAccessKey"
20 });
21
22 //Create connection to database
23 try{
24     //var db = new sqlite3.Database('./db/sensor_logger.db');
25     var db = new sqlite3.Database('/home/pi/Documents/Personal Projects/
26     Cloud Sensors Data Logger/SQLITE_DB/Sensors.db')
27 }catch (err){
28     console.log("Could not connect to database")
29 }
```

```

30 //Create DynamoDb service instance
31 var dynamodb = new AWS.DynamoDB(); // used for real AWS account
32 var docClient = new AWS.DynamoDB.DocumentClient();
33 var table = "SensorReadings";
34
35 var stmt = "SELECT * FROM " + table;
36
37 db.each(stmt, function(err, row){
38     console.log(row.Id, row.Sensor_No, row.Sensor_Type, row.Temperature, row
39     .Humidity, row.Pressure, row.Datestamp);
40
41     var params = {
42         TableName: table,
43         Item:{
44             "Id": row.Id,
45             "SensorNumber": row.Sensor_No,
46             "Type": String(row.Sensor_Type),
47             "Temperature": String(row.Temperature),
48             "Pressure": String(row.Pressure),
49             "Humidity": String(row.Humidity),
50             "Datestamp": row.Datestamp.toLocaleString(),
51         },
52         "ConditionExpression": "attribute_not_exists(Id) and
53             attribute_not_exists(SensorNumber)"
54     };
55
56     docClient.put(params, function(err, data){
57         if (err)
58             console.log(JSON.stringify(err, null, 2));
59         else
60             console.log(JSON.stringify(data, null, 2));
61     });
62
63     console.log("Sensor value with id = " + row.Id + " has been processed
64     and added successfully");
65 });
66 db.close();

```

Listing A.5: Serviciu REST app.JS

A. Fragmente cod sursă

```
1  /**
2   * Created by xps on 11-Mar-17.
3   */
4
5 var AWS = require("aws-sdk");
6
7 AWS.config.update({
8     region: "eu-central-1",
9     endpoint: "http://192.168.0.100:8000/",
10    // accessKeyId default can be used while using the downloadable version
11    // For security reasons, do not store AWS Credentials in your files. Use
12    // Amazon Cognito instead.
13    accessKeyId: "fakeMyKeyId",
14    // secretAccessKey default can be used while using the downloadable
15    // version of DynamoDB.
16    // For security reasons, do not store AWS Credentials in your files. Use
17    // Amazon Cognito instead.
18    secretAccessKey: "fakeSecretAccessKey"
19 });
20
21 console.log("Connecting to localhost port 8000 ...");
22 var dynamodb = new AWS.DynamoDB();
23
24 var params = {
25     TableName : "SensorReadings",
26     KeySchema:[
27         { AttributeName: "Id", KeyType: "HASH"}, //Partition key
28         { AttributeName: "SensorNumber", KeyType: "RANGE" } //Sort key
29     ],
30     AttributeDefinitions: [
31         { AttributeName: "Id", AttributeType: "N" },
32         { AttributeName: "SensorNumber", AttributeType: "N" }
33     ],
34     ProvisionedThroughput: {
35         ReadCapacityUnits: 10,
36         WriteCapacityUnits: 10
37     }
38 };
39
40 dynamodb.createTable(params, function(err, data) {
41     if (err)
42         console.log(JSON.stringify(err, null, 2));
43     else
44         console.log(JSON.stringify(data, null, 2));
45 });


```

Listing A.6: createDynamoDBTable.JS