Rezumatul tezei:

Universitatea POLITEHNICA din București

Facultatea de Automatică și Calculatoare
Departamentul de Calculatoare

TEZĂ DE DOCTORAT

Confidențialitatea datelor în infrastructuri critice

Conducător științific:
Prof. Dr. Ing. Răzvan-Victor Rughiniș

Autor:
Ing. Ioan-Mihail Stan

București, 2022

Abstract

Odată cu lărgirea spectrului de servicii online şi migrarea masivă a maselor, voluntară sau forţată de contexte globale majore, către o interacţiune digitală, perspectiva confidenţialităţii datelor are o nouă dimensiune. De la analiza de date simplă, în scopuri comerciale, la manipulare pentru a destabiliza comunităţile, datele au devenit o formă de exercitare a puterii. Astfel, conştientizarea aspectelor legate de protecţia împotriva expunerii în mediul online este obligatorie.

Conceptul de intimitate presupune conservarea mediului personal şi exercitarea dreptului de a nu fi invadat, în spaţiul personal, de persoane neautorizate. În mediul online, stabilirea limitelor de confidenţialitate este modalitatea de a gestiona datele personale, private, cu scopul de a nu fi expuse terţilor fără un acord prealabil, deliberat. În plus, pentru a scoate în evidenţă, mai bine, problematica din perspectivă tehnică, este esenţial să se exprime noţiunea de confidenţialitate într-un context complex, susceptibil de a fi luat în vizor, datorită relevanţei şi dimensiunii sale, de entităţi rău intenţionate. Astfel, purtarea unei discuţii a problemele de protecţie a datelor în infrastructurile critice devine relevantă. Infrastructurile critice sunt acele infrastructuri considerate esenţiale de către un stat sau o formă de guvernare pentru gestionarea optimă a vieţii cetăţenilor. În general, din perspectiva securităţii statului, primele industrii sau sectoare care devin ţinte, în cazul unui război sau al unui atac cibernetic masiv, sunt sistemele critice.

Teza curentă conturează înţelegerea aspectelor privind confidenţialitatea din cinci puncte de observaţie. Prima este reprezentată de accesibilitatea datelor ca formă de prezentare şi, de asemenea, ca formă de livrare. Aspecte precum ciclul de viaţă al dezvoltării software şi automatizarea sunt studiate pentru a stabili mecanisme de reacţie rapidă la vulnerabilităţi sau puncte de exploatare. Al doilea este anonimitatea datelor expunând metodele şi metodologiile de segregare a identităţii reale de cea din spaţiul digital. Sunt analizate metodele de ascundere a identităţii din tehnologiile blockchain, modelele existente fiind completate cu etape intermediare. Structura discuţiei ştiinţifice se învârte în jurul dezvoltării unui sistem naţional de vot bazat pe soluţii blockchain cu autorizare. Al treilea este legat de transmisia de date în care tehnologiile blockchain sunt luate în considerare pentru capacitatea lor intrinsecă de a furniza topologii de comunicare. Calităţile şi aspectele problematice ale funcţiei de comunicare sunt analizate şi comparate cu paradigma de cloud computing. Al patrulea este legat de distribuţia şi orchestrarea datelor în care este validată fezabilitatea utilizării infrastructurilor containerizate. Este prezentată taxonomia de izolare şi sunt analizate metodele de programare pentru serviciile de date containerizate. Se poate vedea cum politicile utilizatorilor pot avea o reprezentare mai bună peste o taxonomie de izolare redusă în complexitate. Mai mult, studiul propune un mix de modele distincte de guvernare a infrastructurilor distribuite (High Performance Computing, Grid şi Cloud Computing) pentru a reuni aspectele calitative ale fiecăreia, inclusiv aspectele de confidenţialitate a datelor gestionate. Al cincilea arată importanţa stabilirii unei evaluări a riscurilor pentru serviciile de date. Se ia în considerare importanţa utilizării infrastructurilor honeypot pentru expunerea activelor dezvoltate, oferind un cadru de dezvoltare pentru astfel de soluţii şi, de asemenea, o metodologie de implementare în cloud care sa permite coexistenta cu producţia legitimă.

Termeni cheie: Securitate, Initimitate/Confidenţialitate, Protecţia datelor, Blockchain, Calcul de înaltă performanţă (HPC), Grid (calcul peste reţea), Container, Cloud, Honeypots, Automatizare

Context

În ultimii ani, comunitatea globală s-a confruntat cu mai multe crize şi evoluţii prospective majore care au dus la o accelerare sporită a transformării digitale. Criza Covid-19 ne-a arătat cum

putem depăși granițele distanțării fizice, mutând întreaga activitate online. S-a înregistrat o creștere majoră a utilizării serviciilor colaborative, a instrumentelor de videoconferință, a soluțiilor VPN [1] și altor unelte, ca urmare a unor reforme majore în sectoare precum educația, tehnologia informației sau serviciile guvernamentale. În timpul izolarii, s-au schimbat și obiceiurile comportamentale ale utilizatorilor. Aceștia au suplimentat o parte din activitatea recreativă, în interiorul mediul online. S-au putut observa creșteri majore în comerțul online, ceea ce a dus ulterior la o criză în lanțul de aprovizionare, generând întârzieri semnificative. În același timp, criza semiconductorilor și imposibilitatea producerii cipuri a făcut ca industriile critice să nu poată livra la timp, afectând sectoare importante, cum ar fi manufacturile critice, transporturile și alte sectoare care se bazează pe sisteme electronice. În același timp, tensiunile post-pandemice au produs conflicte și războaie care au destabilizat sectoarele financiar și energetic, în special în spațiul european, prin conflictul dintre Rusia și Ucraina. Nu în ultimul rând, votat în 2016 și implementat în 2018, Regulamentul general privind protecția datelor, a pus acele companii care gestionează datele personale ale cetățenilor Uniunii Europene, într-o continuă restructurare a infrastructurii IT, pentru a acoperi noile prevederi critice. După cum se poate observa, astfel de situații majore implică un stres suplimentar și generează premisele unui test de anduranță în unele dintre sectoarele industriale cheie.

Odată cu migrarea masivă a activității zilnice online, s-a putut observa, de asemenea, o creștere semnificativă a atacurilor în rețea și a expunerii utilizatorilor în spațiul online [2]. Astfel, problema securității și, în special, a confidențialității a atins noi culmi, observându-se o creștere a metodelor de atac, a aplicațiilor exploatabile, dar încă intens utilizate și a ratelor de succes în compromiterea bunurilor publice. Devine din ce în ce mai relevantă identificarea și înțelegerea, în detaliu, a oportunităților de exploatare și a contramăsurilor. Toate mecanismele de apărare se pot baza pe metodologii generale și standarde industriale sau pot fi personalizate pentru fiecare tip de active expuse. Devine relevant să vorbim despre utilizarea pe scară largă a infrastructurilor de tip honeypots[1] ca modalitate de a genera rapoarte de securitate pentru fiecare serviciu expus, despre orchestrarea inteligentă a aplicațiilor care acoperă necesarul de păstrare a confidențialității datelor expuse, despre comunicarea anonimă și transferul securizat al activelor digitale prin intermediul unor tehnologii precum blockchain[2], despre partajarea infrastructurilor mari între mai mulți chiriași/utilizatori cu scopul de a facilita transformarea digitală și așa mai departe.

Teza acoperă multiple aspecte ale prezervării confidențialității în diverse infrastructuri. Ea cuprinde atât metode și metodologii, propune soluții, analizează situații critice, observă oportunitățile de transformare digitală, sugerând forme prin care tranziția se poate face cu un efort minim necesar pentru a acoperi nevoile relevante de securitate și confidențialitate pentru gestionarea datelor critice, precum și alte aspecte relevante. Punctul central este pus pe infrastructurile critice, acele infrastructuri costisitoare, sensibile la crize globale sau locale și care, în contextul incapacității de a furniza servicii, afectează masele. Infrastructurile critice digitale tind să implementeze paradigma multi-tenancy[3] și se bazează în principal pe distribuția efortului de calcul până la descentralizare. Astfel, analiza științifică va izola și lua în considerare în principal următoarele atribute în ceea ce privește definirea infrastructurii critice: implementări masive, infrastructuri distribuite, centralizate sau descentralizate, dispersate geografic sau menținute în centre de date mari, capabile să deservească mai mulți chiriași/utilizatori.

[1] borcan cu miere - terminologie utilizată pentru a descrie implementările de infrastructuri menite să atragă atacatori cibernetici spre a fi analizati într-un context similar producției

[2] lanț de blocuri - terminologie care definește o categorie de tehnologii din piață

[3] multi-chiriaș - terminologie care sugerează posibilitatea delimitării unor spații de lucru izolate pentru utilizatori multiplii

Motivație și obiective

Păstrarea confidențialității datelor este o problemă reală cu impact global. Înțelegerea oportunităților și a metodelor prin care poate fi garantată protecția datelor în infrastructurile mici, medii și mari reprezintă o preocupare relevantă în activitatea inginerilor IT, deoarece aceștia sunt adesea puși în situația de a reacționa rapid la orice eveniment perturbator și de a identifica potențialele puncte vulnerabile. Astfel, teza curentă analizează contextul global din perspectiva infrastructurilor complexe și costisitoare care trebuie să asigure un nivel adecvat de confidențialitate și securitate a datelor. Subiectele studiate acoperă o gamă largă de puncte de interes în jurul păstrării confidențialității datelor:

- virtualizarea ușoară cu containere pentru izolarea contextului critic de procesare a datelor
- orchestrarea infrastructurilor mari, containerizate, pentru a asigura protecția și coerența datelor încă din etapa de implementare/lansare în context de producție
- blockchain pentru aportul său în comunicarea securizată și anonimă și, de asemenea, pentru funcția de stocare adecvată oferită în contextul datele critice
- honeypots ca modalitate de stabilire a parametrilor organici pentru măsurarea nivelului de securitate și confidențialitate al serviciilor expuse
- sisteme distribuite de mari dimensiuni, cum ar fi grid computing (procesare în rețele extinse) sau calculul de înaltă performanță, centralizate sau descentralizate, și politicile pe care acestea le aplică pentru a menține securitatea, confidențialitatea și coerența datelor
- automatizarea pentru a asigura o reacție rapidă la evenimente neașteptate, exploatări malițioase de date sau atacuri cibernetice
- metode și metodologii pentru o transformare digitală rapidă, fără a pierde din vedere securitatea și confidențialitatea datelor gestionate

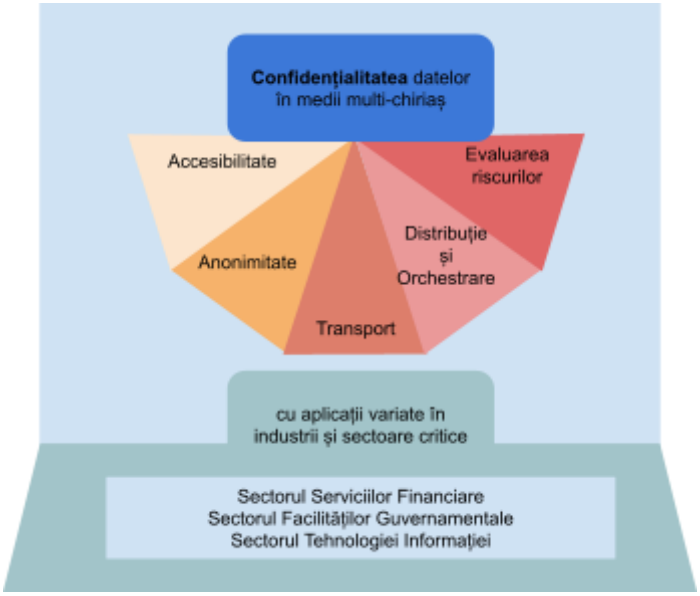

Figura 1 - Obiectivele tezei și studiile de caz

Obiectivele tezei sunt ilustrate în Figura 1, care propune cinci dimensiuni prin care este analizată asigurarea confidențialității datelor. Aceste infrastructuri aflate sub observație sunt infrastructuri critice, propunând trei exemple formale de sectoare industriale pe care am construit studii de caz. Cu toate acestea, teza abordează un domeniu mai larg de aplicabilitate, studiile generate fiind valabile și portabile și la infrastructuri non-critice de dimensiuni variabile. Cele cinci dimensiuni propuse sunt:

- Accesibilitatea, care definește metode și metodologii de expunere a datelor pentru a fi consumate, propune automatizarea ciclurilor de dezvoltare și de lansare, propune metode de prezentare a serviciilor de date în infrastructuri partajate.
- Anonimitate, care evidențiază formele de stabilire a separării identităților reale și digitale, fără a fi necesară obstrucționarea participării la acțiuni esențiale, cum ar fi procesele civice sau guvernamentale, în care este necesară o autentificare coerentă, a priori.
- Transportul, care prezintă metode de menținere a confidențialității și consecvenței datelor în timp ce acestea sunt deplasate sau sunt tranzacționate ca informații private, între entități participante; expune limitările facilitatorilor de comunicare.
- Distribuția și orchestrarea, care prezintă mecanisme ce realizează izolarea și împachetarea datelor ca parte a efortului de asigurare a secretizării și a portabilității acestora; metode de distribuire sau localizare a datelor; metode de programare și de lansare în execuție a procesatorilor de date, în apropierea depozitelor de date pentru a evita eventualele întârzieri generate de migrarea acestora; orchestratori care garantează aplicarea corectă a politicilor de calitate a serviciului sau a politicilor de utilizare; forme de întrepătrundere a unor orchestratori din zona infrastructurilor distribuite menite să pună împreună punctele tari ale fiecăruia cu scopul de a acoperi anumite scenarii de manipulare a datelor.
- Evaluarea riscurilor, care oferă metode și metodologii pentru construirea unor rapoarte coerente de securitate și confidențialitate pentru bunurile și serviciile care urmează să fie expuse în exterior; metode și metodologii pentru construirea unor infrastructuri menite să atragă utilizatorii rău intenționați pentru analiza organică a modalităților lor de a ocoli restricțiile.

Contribuțiile de cercetare

Teza include atât studii de caz, cât și soluții practice în domeniul securității și confidențialității sistemelor complexe și critice, inclusiv adoptarea de tehnologii, metodologii și practici emergente. Prin urmare, în cadrul studiului am adoptat și adaptat tehnologii precum Kubernetes, Blockchain, motoare de containerizare sau paradigme precum migrarea spre cloud[4], calcul în rețele grid și calculul de înaltă performanță sau tranziția către arhitecturi pe microservicii. Principalele contribuții sunt orientate spre o prezentarea generală a unor arhitecturi și unor metodologii de arhitecturare pentru sisteme complexe digitale menite să depășească provocările majore legate de securitate și confidențialitate, în contextul unor infrastructuri complexe, multi-nod, multi-chiriaș, scalabile, care acoperă o gamă largă de scenarii de utilizare: de la servicii guvernamentale la servicii financiar sau servicii din sfera tehnologiei informației. Printre contribuțiile mele se numără următoarele:

- a fost propusă o metodologie care să susțină treptat tranziția soluțiilor complexe de orchestrare a calculatoarelor de înaltă performanță, concepute ca monolite, către paradigma microserviciilor
- a fost implementat un flux de lucru automatizat în contextul uneia dintre cele mai scumpe, critice și importante infrastructuri de calcul de înaltă performanță și de calcul în rețea, găzduită de experimentul ATLAS de la CERN, care permite chiriașilor să vizualizeze și să manipuleze în mod corespunzător rezultatele provenite din procesarea datelor în topologii de învățare automată, la cerere
- a fost furnizată o analiză complexă privind modul de adaptare a Kubernetes pentru a susține volumul de lucru în calcul de înaltă performanță, punând accentul pe menținerea contextului utilizatorului, izolat și securizat

[4] nor - terminologie care definește un ansamblu distribuit de servicii digitale

- a fost generată o taxonomie de clasificare pentru adaptările Kubernetes în calcul de înaltă performanță pentru literatura științifică recentă
- a fost prezentate îmbunătățiri simple, dar importante, ale adaptărilor Kubernetes în calcul de înaltă performanță din literatura științifică recentă, pentru a susține mai bine modelul propus și metodele adoptate
- a fost condus un studiu privind securitatea și confidențialitatea motoarelor de containerizare disponibile pe piață, cu accent pe problema escaladării privilegiilor, investigând modul în care, prin simplificarea taxonomiei de izolare, se poate crește rezistența la atacurile cibernetice
- a fost propus un concept de arhitectură pentru un sistem pe Kubernetes care poate constitui baza pentru proiectarea unui mecanism de guvernanță HPC-Grid (calcul de înaltă performanță - calcul peste rețea), urmând modelele și practicile moștenite din paradigma cloud.
- a fost rezentată o metodologie de proiectare a unui sistem de alegeri electronice prin Blockchain, cu accent pe confidențialitate și anonimizare, prin aplicarea unui model de abstractizare a datelor în trei etape.
- a fost propusă o metodologie complexă pentru a ajuta arhitecții să definească arhitecturi de tip honeypots peste Kubernetes în coexistență cu mediul de producție legitim.
- a fost furnizată o euristică și un cadru de lucru pentru definirea generatorilor de honeypot-uri pentru a ascunde posibilele similitudini între instalari succesive; a fost furnizată o euristică și o arhitectură pentru definirea generatorilor de honeypot-uri pentru a face astfel de sisteme critice rezistente la identificarea sabloanelor/matrițelor de instalare la construcția lor în infrastructuri distincte
- a fost furnizată o euristică care implementează tranzacții în timp real pe soluții blockchain publice lente

1. Accesibilitatea datelor

Primul punct care trebuie definit în ceea ce privește confidențialitatea și securitatea datelor este accesibilitatea datelor. Prin urmare, este esențial să se înțeleagă cateva metodologii de expunere a datelor către lumea exterioară și mecanismele care facilitează o reacție rapidă în caz unei defecțiuni a serviciilor din spate sau în contextul unei actualizări a aplicației consumabile cu constrângeri de înaltă disponibilitate. Astfel, explorez capacitățile unui mediu cloud ce poate fi utilizat pentru expunerea datelor critice și, totodată, oportunitățile de extindere a infrastructurilor de procesare distribuită existente, cu funcții și servicii emergente. Scopul este acela de a asigura un efort facil de modificare a codului acestor sisteme pentru a permite introducerea de noi funcții în infrastructuri critice, costisitoare, vechi și îndelung dezvoltate, cu scopul de a atribui politicile moderne de protecție a datelor.

1.1. Întrebări de cercetare

- Care sunt oportunitățile de a aborda automatizarea necesară pe parcursul ciclului de viață al dezvoltării de software în infrastructurile critice?
    - Cum se poate realiza o detecție timpurie a potențialelor puncte exploatabile în expunerii datelor?

- Cum se poate permite o reacție rapidă în cazul unei exploatări de tip ziua 0[5]?

1.2. Contribuții

- Stan, Ioan-Mihail, Siarhei Padolski, and Christopher Jon Lee. "Exploring the self-service model to visualize the results of the ATLAS Machine Learning analysis jobs in BigPanDA with OpenShift OKD3." *EPJ Web of Conferences*. Vol. 251. EDP Sciences, 2021.

1.3. Fundament tehnic

1.3.1. DevOps și DevSecOps

Figura 2 - Ciclul de dezvoltare de software - Fluxul de date

O bună oportunitate de a susține efortul de transformare digitală pentru a moderniza soluțiile software existente este prin externalizarea unor funcții mici, dar relevante, către un furnizor de cloud. Prin urmare, în loc de a insera întreaga logică a unei noi funcții într-o bază de cod imensă, definind uneori structuri monolitice și complexe, o modalitate de a îmbunătăți procesul de dezvoltare este de a le detașa de soluția principală și de a le proiecta ca servicii individuale sau chiar microservicii. Prin urmare, efortul de a răspunde la cererile de modificare se reduce doar la definirea unor rutine la distanță slab cuplate și nu la încorporarea întregii logici noi în structura complexă a sistemului. Pentru a susține o creștere semnificativă a efortului în modernizarea infrastructurii moștenite, cu un astfel de model de gestiune, este necesar să se impună o mentalitate spre automatizare în rândul celor care contribuie la tranziția digitală. Un pas important este adoptarea metodologiilor și practicilor din cultura DevOps și generarea de linii de producție automatizate (linii de asamblare) care să urmeze fluxul de date al ciclului de viață al dezvoltării de software (Figura 2). O practică importantă, esențială pentru dezvoltarea modernă de software este integrarea continuă (eng. continuous integration) [45]. Prin definiție, integrarea continuă acoperă procesul de livrare a soluțiilor software de la nivel cod până la instalarea în infrastructura de testare a integrării (Figura 3).

[5] recent descoperit

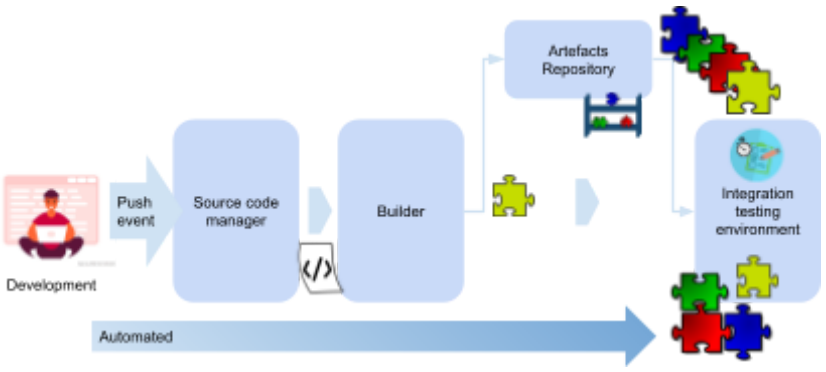

Figure 3 - Continuous Integration (Integrare continuă)

O poartă de calitate poate fi stabilită în momentul publicării codului, putând fi realizată prin efectuarea unei analize statice a codului înainte de a merge mai departe și de a declanșa etapa de construcție a elementului executabil. În acest punct, o analiză statică a codului poate identifica fragmente anti șablon din structura software și poate propune standarde de dezvoltare, cu scopul de a îmbunătăți nu numai eficiența, ci mai ales securitatea și confidențialitatea datelor expuse. De exemplu, așa cum P. Ferrara et al. prezintă în [46], într-o astfel de etapă timpurie, se poate stabili o vizualizare coerentă asupra cerințelor de confidențialitate asupra politicii GDPR. O data ce se trece de poarta de calitate, se poate declanșa etapa de construire a artefactelor executabile, care, în unele cazuri, necesită alocarea de resurse fizice. Odată cu evoluția sistemelor de containerizare și cu adoptarea în masă a paradigmei cloud, alte artefacte compilabile pot fi reprezentate de imaginile de container. Containerele, pentru capacitatea lor de a rula în medii eterogene și pentru portabilitatea lor intrinsecă, reprezintă o alternativă pentru livrarea de software, foarte asemănătoare unui sistem clasic de gestionare a pachetelor.

Odată construite, toate artefactele (binare sau imagini de containere) trebuie să fie stocate în depozite de artefacte dedicate. De aici, etapele secvențiale vor descărca obiectele executabile și le vor plasa în contexte de funcționare. În plus, etapa de construire poate integra o altă poartă de consistență/calitate reprezentată prin execuția testelor unitare. Dacă testele unitare sunt executate cu succes, componenta software este calificată pentru a trece mai departe la următoarea etapă din ciclul de dezvoltare.

Ultima etapă a practicii de integrare continuă este testarea în integrare. Aici, dacă face parte dintr-o soluție mai mare, fiecare obiect software, construit anterior, este testat împreună cu toate celelalte aplicații sau servicii pentru a identifica orice problemă de participare în cadrul ansamblului. Astfel, în timp ce în etapele anterioare, artefactele erau testate individual, acum sunt validate împreună ca un întreg. La finalul acestei sesiuni, se generează un livrabil care trebuie să treacă printr-o etapă de testare de acceptare, înainte de a fi pregătit pentru a fi lansat într-un context de producție. În toate nivelurile intermediare, o parte importantă a validării maturității fiecărui artefact constă în verificarea acestora din mai multe unghiuri. Prin urmare, o inserție importantă între aceste etape obișnuite este o testare riguroasă a securității și a confidențialității. Aici accentul trebuie pus pe scurgerile de date, escaladarea privilegiilor, vulnerabilitățile din bibliotecile utilizate și așa mai departe. Prin urmare, ca o măsură a maturității, soluția software trebuie să fie rezistentă la cel puțin vulnerabilitățile și metodele de atac bine cunoscute. înainte de a fi lansată.

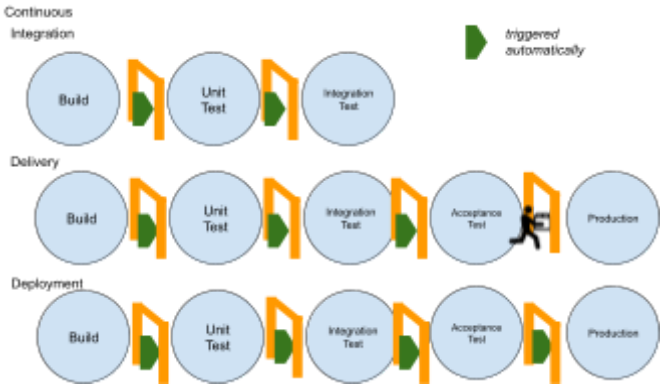

Figura 4 - Continuous Integration, Continuous Delivery, Continuous Deployment

Odată ce etapa de acceptare este îndeplinită, trebuie să se decidă dacă soluțiile dezvoltate pot fi lansate automat în producție sau necesită intervenție umană (Figura 4). Această graniță subțire între cele două abordări delimitează două practici comune, și anume livrarea continuă (continuous delivery) și implementarea continuă (continuous deployment). Prima se bazează pe realizarea continuă de produse viabile, în timp ce cealaltă își asumă riscul unui flux continuu de dezvoltare și lansează în producție fiecare modificare care a trecut de porțile de calitate stabilite. Această ultimă etapă este, de asemenea, evaluată în mod automat. Termenul continuu implică faptul că, în toate etapele vizate, tranziția trebuie să se facă în mod automat.

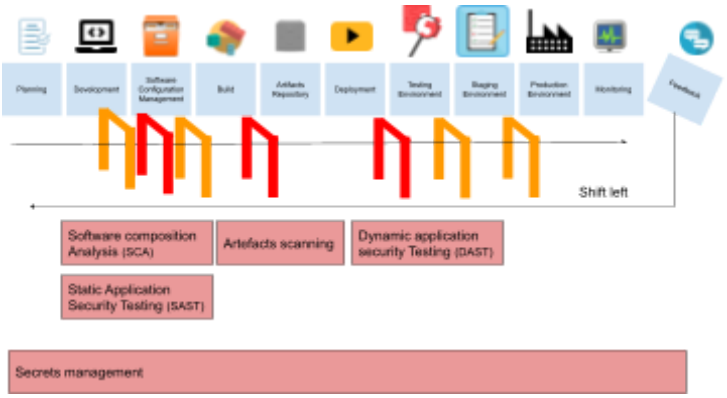

Figura 5 - Tranziția de la DevOps către DevSecOps

Ca și în cazul DevOps [52], DevSecOps este mai mult o tendință culturală în industria IT, susținută de instrumente, automatizare și mentalitate de învățare și îmbunătățire continuă a fluxurilor utilizate la nivel intern ca parte a ciclului de viață al dezvoltării de software. Din punct de vedere pragmatic, DevSecOps impune o deplasare spre stânga a validărilor de securitate (și de confidențialitate) cu scopul de a depăși blocajul generat de echipele de securitate în timpul evaluării produselor livrabile înainte de a le trece în producție. După cum se poate observa în Figura 5 și după cum a fost definit de R. Kumar [53] et al., se pot efectua analize specifice de securitate și confidențialitate în fiecare etapă a ciclului de viață. Astfel, echipele de securitate trebuie să facă o selecție a instrumentelor automate, să stabilească constrângerile și politicile de securitate și de confidențialitate a datelor dorite în contextul livrabilelor produse și să instruiască dezvoltatorii și echipele operaționale în legătură cu principiile de lucru. O astfel de abordare multidimensională accelerează timpul de lansare pe piață și oferă premisele producerii de software în flux continuu, cu livrabile care sunt prezentate într-o formă matură și consumabilă. Dezvoltarea de software este alimentată de diverse etape de analiză a securității și confidențialității: Analiza compoziției sursei

(SCA), Testarea statică a securității aplicațiilor (SAST), Testarea dinamică a securității aplicațiilor, Managementul secretelor etc., în timp ce echipele operaționale se concentrează pe întărirea infrastructurii de bază, securitate contextului de execuție, monitorizare și așa mai departe [53].

1.3.2. Infrastructura ATLAS de la CERN

Fiecare dintre principiile și practicile prezentate în secțiunea curentă au fost luate în considerare în elaborarea unui studiu de fezabilitate privind transformarea digitală și extinderea infrastructurii de calcul distribuit a experimentului ATLAS de la CERN. Efortul depus a fost acela de a integra noi funcții prin intermediul paradigmei cloud cu un efort minim necesar în ceea ce privește modificarea bazei de cod complexe construite de-a lungul mai multor ani de existență. Accentul a fost pus pe propunerea de noi modele arhitecturale, cum ar fi principiul serviciilor distribuite și al microserviciilor, și pe realizarea automatizării pe tot parcursul ciclului de dezvoltare a noilor funcții. Soluția propusă a fost creată pentru a sprijini tranziția către principiile și mentalitatea DevSecOps, în special din perspectiva criticității infrastructurii și a importanței acesteia într-un context global. Experimentul a fost construit în jurul unei soluții de vizualizare a datelor procesate în infrastructura HPC și Grid(calcul de înaltă performanță - calcul peste rețea) pentru sarcini de învățare automată, deoarece, într-un context intern, acestea necesită o gestiune specială și o metodologie de manipulare dedicată, cu scopul de a asigura confidențialitatea datelor și calitatea multi-chiriaș a infrastructurii. Facilitatorul de soluții cloud a fost OKD versiunea 3 (o variantă Kubernetes pentru întreprinderi, dezvoltată de comunitatea RedHat OpenShift).

Înainte de a merge mai departe, voi prezenta pe scurt structura infrastructurii de calcul ATLAS dedicată simulărilor și proceselor de analiză.

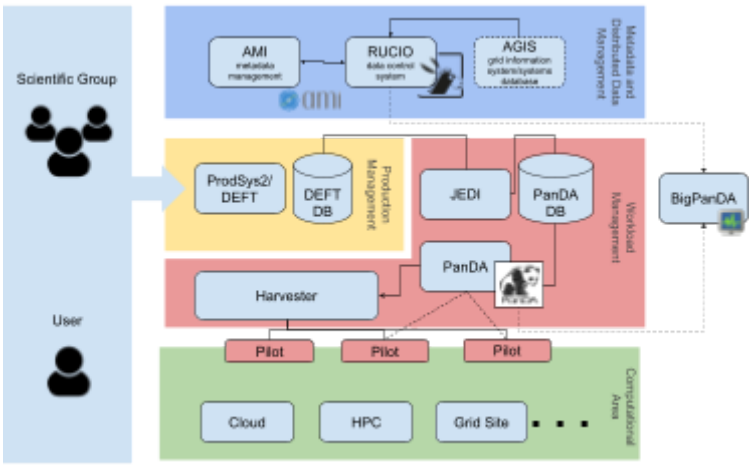

Figura 6 - Sistemul de calcul distribuit ATLAS (Infrastructura de analiză)

Sistemul de calcul distribuit ATLAS oferă grupurilor științifice și membrilor capacitatea de a analiza colecții uriașe și scumpe de date, generate ca efect al coliziunile de particule în Large Hadron Collider (HLC) și detectate ca evenimente în senzorul ATLAS de la CERN [55]. În același timp, pentru validarea raționamentelor științifice, aceeași infrastructură servește la rularea simulărilor Monte Carlo [56] și, de asemenea, la executarea unor scenarii de producție distincte, provenite din alte domenii științifice. După cum se poate observa în Figura 6, compilată din informațiile și șabloanele arhitecturale prezente în [57][58][59], infrastructura abstractizează și orchestrează cererile de executare a proceselor științifice, ascunzându-se în spatele unui sistem complex de gestionare a datelor, metadatelor și a sarcinilor de lucru. Ansamblul este capabil să ruleze pe sisteme eterogene,

distribuite pe zone geografice mari și să implementeze diferite paradigme (HPC și supercomputere, cloud, grid, rețele universitare etc.). Sistemul, în sine, îmbrățișează paradigma Grid Computing, adaptată pentru a sprijini varietatea de sisteme implicate în cercetarea științifică. Orchestratorul sarcinilor/lucrărilor de analiză sau de simulare primește diverse intrări pentru poziționarea și executarea optimă a acestora, incluzând aici problema localizării datelor, care pot fi în cantități uriașe, greu de mutat în apropierea unui motor computational. Alte probleme evidente sunt cele legate de disponibilitatea resurselor sau de constrângerile de execuție și de politicile de utilizare care se aplică grupurilor științifice or indivizilor.

Creierul infrastructurii de calcul ATLAS este managerul de fluxuri de lucru PanDA [58], care reunește în jurul său toate sistemele esențiale. Acesta implementează sistemul de grupare în loturi, preluând sarcinile standardizate de la platformele de submisie și plasându-le în cozi de prioritate. Pe baza datelor de intrare de la furnizorii de metadate, acesta comunică cu sistemul tampon Harvester sau direct cu sistemele computaționale Pilot pentru a transmite sarcinile/lucrările care urmează să fie executate de motoarele de procesare. PanDA este echipat cu un sistem de monitorizare a sarcinilor/lucrărilor pe toată durata lor de viață, ceea ce îl face un candidat perfect pentru proiectul propus de mine. Platforma de monitorizare se numește BigPanDA [59].

1.4. Studiu de caz - Serviciu cloud de vizualizare a datelor cu suport multi-chiriaș pentru calcul de înaltă performanță și procesare Grid (peste rețea)

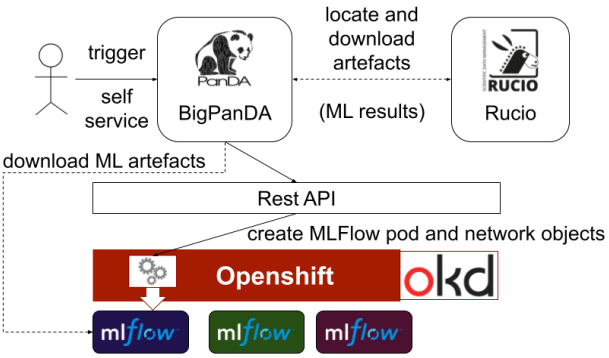

Figura 7 - Arhitectura concept

În studiul de caz actual [60], mă concentrez pe definirea interacțiunii dintre BigPanDA și OpenShift OKD. Contextul unei astfel de interacțiuni este legat de necesitatea de a vizualiza datele de analiză provenite din învățarea automată (Machine Learning) într-un format prietenos. În același timp, se testează un model de gestionare a acestor fluxuri de date, detașat de centrul de orchestrare de bază al infrastructurii de calcul ATLAS. După cum se poate observa în arhitectura conceptului și în fluxul de interacțiune (Figura 7), un utilizator, care a trimis anterior o sarcină de învățare automată către infrastructura de procesare ATLAS, poate solicita, de asemenea, un serviciu de vizualizare pentru a afișa rezultatele. Pentru fiecare cerere, centrul de control BigPanDA declanșează crearea unui serviciu web în OpenShift OKD. Ca parte a acestei rutine, rolul BigPanDA este de a localiza rezultatele și de a le descărca[59] din infrastructura distribuită ATLAS, prin intermediul Rucio[72]. Odată ce datele sunt stocate și indexate în BigPanDA, centrul de control apelează intefața API OpenShift OKD și

declanşează crearea unui serviciu web MLFlow împreună cu toate obiectele de comunicare şi configurare OpenShift (Kubernetes). Podul MLFlow descarcă artefactele de învățare automată din BigPanDA şi le stochează local, într-o locație temporară/volatilă.

Urmând o astfel de abordare, soluția BigPanDA externalizează gestionarea vizualizării datelor către o platformă cloud externă, optimizată pentru acest tip de interacțiune. Principiul pe care am bazat modelul arhitectural este "segregarea sarcinilor". Prin urmare, în loc de a avea o singură soluție care să se potrivească tuturor fluxurilor, diverse rutine auxiliare pot fi detașate de soluția principală şi executate prin intermediul unor platforme specializate. OpenShift OKD este o platformă viabilă pentru diverse scenarii, în special atunci când este vorba de crearea de servicii la cerere, de izolarea de tip multi-chiriaş şi de gestionarea inteligentă a containerelor. Pentru a profita de OpenShift şi, de asemenea, pentru a creşte portabilitatea soluției mele, livrez instanțele MLFlow în containere şi poduri. Urmând acest model nativ de cloud, soluția este extrem de portabilă şi poate fi ajustată cu uşurință pentru a rula pe majoritatea implementărilor cloud publice şi private.

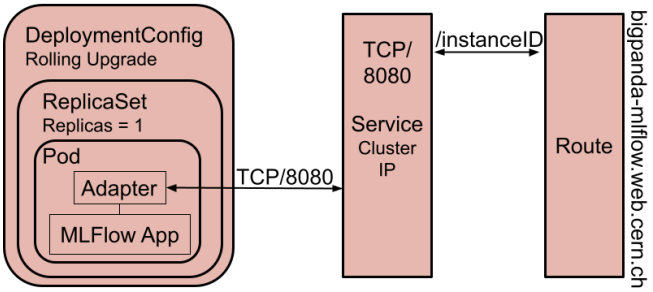

Figura 8 - Modelul de comunicare şi obiectele OpenShift

OpenShift OKD este o implementare Kubernetes, prin urmare, moşteneşte aceeaşi arhitectură de bază ca şi cea implementată de proiectul original. Un serviciu care trebuie să fie expus lumii exterioare va fi accesat cel mai adesea prin intermediul sistemului de echilibrare a încărcării cu sarcini (load-balancer), poziționat la intrarea în infrastructură. Prin urmare, dacă un client ar dori să se conecteze la un serviciu care rulează în interiorul unui cluster OpenShift OKD, clientul va putea utiliza un nume de domeniu specific pentru a ajunge la serviciul respectiv. Mai mult, un controler de intrare poate, de asemenea, să manipuleze rutele interne pentru a redirecționa traficul către diverse servicii pe baza unor subdomenii sau a unor selectori de căi (URL-uri specifice). BigPanDA va utiliza mecanismul de rutare (controlerul de intrare) pentru a facilita conceptul de multi-chiriaş prin crearea unor definiții unice de tip fan-out (Figura 8) şi a unor primitive de comunicare pentru fiecare instanță MLFlow, prin intermediul OKD API.

În plus, am observat că atât OKD 3.11, cât şi MLFlow v1.9 (versiunile selectate pentru realizarea proiectului pilot) nu acceptă rescrierea țintelor (target rewriting). Acest concept înseamnă că, în cazul în care o cale de solicitare HTTP nu corespunde unei resurse expuse, solicitarea va sfârşi prin a genera o eroare de tip "resource-not-found". Deoarece voi utiliza un singur domeniu definit în serverul de nume (DNS), pentru toate instanțele MLFlow care rulează în paralel, acestea se pot identifica printr-un şir aleatoriu încorporat în calea resursei (URL). Unele servere web sau aplicații web acceptă redirecționarea unei cai inexistente către calea rădăcină. Cu toate acestea, atât comportamentul de baza OKD 3.11 cu HAProxy Ingress Controller ataşat, cât şi MLFlow v1.9 nu au suport pentru o astfel de configurare. Pentru a rezolva această problemă, am adoptat un model de proiectare cu mai multe containere pentru construcția poduri - modelul adaptor[73]. Un obiect adaptor este situat în fața serviciului principal al aplicației MLFlow şi gestionează cererile venite de la

load-balancer. Fiecare cerere HTTP va fi tradusă și trimisă serviciului MLFlow prin intermediul interfeței localhost (Figura 8).

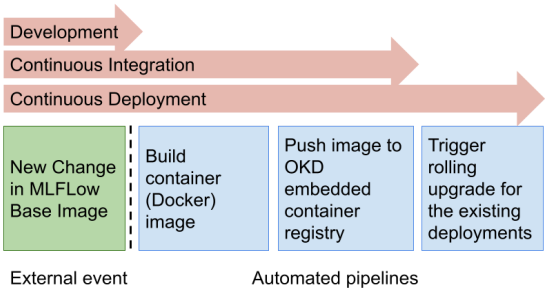

Figura 9 - Practici DevOps

Tot ca parte esențială a viziunii arhitecturale, am decis să creez propria imagine de container de bază. Acesta include aplicația suport MLFlow și scripturile de asamblare și este generat folosind capacitățile de construire/compilare native disponibile în OpenShift OKD. Prin urmare, am separat pregătirea imaginii MLFlow de artefactele de definirea a fluxului de instalare efectivă a aplicației, respectând metodologia și practicile DevOps (Figura 9). În prezent, toate elementele de configurare sunt stocate separat de baza de cod BigPanDA. În momentul în care este detectată o modificare în baza de date de gestiune a codului, un declanșator execută fluxul de lansare continuă a artefactelor (Continuous Deployment pipeline), construind imaginea de baza, pregatind aplicația suport și lansand serviciul de vizualizare în producție.

1.5. Prezentare generala a perspectivelor

În acest studiu de caz am dezvoltat o nouă funcție pentru BigPanDA care oferă posibilitatea de a vizualiza rezultatele învățării automate, produse în infrastructura de calcul distribuit ATLAS. Un chiriaș (doctorand, grup științific etc.) poate solicita un astfel de serviciu direct din platforma de monitorizare BigPanDA, iar efortul computational va fi delegat, în continuare, unei implementări OpenShift OKD prin intermediul executarii unor rutine la distanță via apeluri de tip REST API. OpenShift va genera un pod MLFlow pentru fiecare solicitare, va descărca artefactele provenite din procesarea de tip învățare automată de la BigPanDA și va expune serviciul web către exterior menținând o disponibilitate ridicată prin intermediul mecanismelor native de vindecare. Urmând acest model de delegare, am demonstrat că BigPanDA poate adopta cu ușurință o abordare arhitecturală nativă cloud și, de asemenea, că poate funcționa ca un catalog de servicii computaționale cu caracter științific, în contextul infrastructurii distribuite de calcul ATLAS de la CERN. În plus, am urmat, de asemenea, mai multe metodologii și practici DevOps pentru a facilita construirea containerului MLFlow de bază de la zero. Prin urmare, am implementat fluxuri de integrare continuă și de lansare în producție continuă utilizând mecanisme native din portofoliul OpenShift OKD. Aceste fluxuri sunt declanșate automat de fiecare dată când apare un eveniment PUSH în depozitul de elemente de configurare externă (stocate sub formă de cod într-un limbaj declarativ). În plus, ca o bună practică, strategia de lansare în producție utilizează modelul de actualizare continuă, o metodă care minimizează timpii de indisponibilitate, inevitabil în derularea proceselor de actualizare. În cele din urmă, am implementat, de asemenea, o soluție de curățare via o procedură Python care identifică instanțele de servicii vechi și le șterge dacă au depășit un termen de expirare stabilit la 24 de ore.

Rezultatele obținute în timpul fazei de testare certifică funcționalitatea completă a soluției integrate. Pentru noi iterații, am identificat deja alte câteva căi de dezvoltare și optimizare și aici includ: utilizarea containerelor de inițializare, o migrare către modelul operatorilor, utilizarea sondelor de asigurare a sănătății aplicației containerizate și delegarea procesului de rutare către o altă soluție de controler de intrare în infrastructura de comunicare. Aceste schimbări pot aduce îmbunătățiri semnificative arhitecturii soluției, deoarece aplică un model de separare a blocurilor funcționale mai performant și, de asemenea, elimină necesitatea de a avea componente suplimentare, cum ar fi un obiect de tip adaptor..

În studiul de caz actual, am prezentat o modalitate de expunere a datelor critice în mediile complexe, vechi și cu mai mulți chiriași prin externalizarea funcției de vizualizare a datelor către un mediu cloud extern. O modalitate ușoară de a restricționa accesul la date, accesibil oricărui dezvoltator, este implementarea unor metode ușoare de ofuscare a coordonatelor de acces. În plus, automatizarea joacă un rol esențial ca metodă de a reacționa rapid în cazul unui pericol major. Toate aceste implementări sunt componente relevante în stabilirea dimensiunii de accesibilitate a datelor.

2. Anonimitatea datelor

În orice schimb de date, infrastructura de transmisie trebuie să se asigure că nu sunt dezvăluite date critice sau private. Aceasta este o funcție obligatorie de securitate și confidențialitate și este frecvent adăugată ca o capacitate nativă a facilitatorilor de comunicații. De la apeluri API HTTPS asincrone între servicii care rulează împreună pentru a furniza rezultate complexe până la criptarea și semnarea în siguranță a datelor înainte de a le stoca într-o bază de date distribuită și descentralizată, multe tehnologii acoperă astfel de nevoi și le oferă fără niciun efort suplimentar, în timpul implementării soluțiilor. Cu toate acestea, astfel de capacități stabilesc doar granițe care îngrădesc informațiile transmise între entitățile finale și nu obturează sursa datelor sau informațiile care vizează entități sau persoane. Astfel, ele asigură doar jumătate din nevoile de confidențialitate atunci când se efectuează analize asupra datelor critice. Cealaltă jumătate trebuie să se ocupe de cunoștințele reale partajate și trebuie să ascundă orice informații private despre originea datelor sau despre publicul vizat [75][76]. Blockchain oferă colegilor mecanisme pentru a-și ascunde identitatea reală în spatele unei identități digitale. Cu toate acestea, după cum prezintă Q. Feng et al. în lucrarea lor privind stadiul actual al tehnologiei [80] , prin natura sa de la egal la egal, o implementare publică dezvoltată în jurul criptomonedelor nu este pe deplin protejată împotriva analizei rețelei sau a analizei de grupare a adreselor, care ar putea dezvălui sursa datelor. Abordarea mea constă în adaptarea și distribuirea efortului de manipulare a datelor către mai multe instituții într-o implementare privată și cu permisiune, în timp ce o parte din euristica și funcțiile de anonimat și ofuscare sunt externalizate către alte tehnologii personalizate, în afara lanțului de blocuri (off-chain). Demonstrez că soluțiile blockchain pot acoperi nevoile unui proces guvernamental critic - procesul electoral de stat, urmărind și agregând poveștile de succes din țările care au derulat anterior astfel de proiecte-pilot de e-alegeri.

2.1. Întrebări de cercetare

- Cum se poate asigura anonimatul/confidențialitatea informațiilor asupra datelor publice[6] în sistemele critice care necesită autentificarea utilizatorului?

[6] partajate/expuse în afara premiselor utilizatorului

2.2. Contribuții

- Stan, Ioan-Mihail, Ilie-Constantin Barac, and Daniel Rosner. "Architecting a scalable e-election system using Blockchain technologies." *2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2021.

2.3. Studiu de caz - Blockchain în servicii guvernamentale

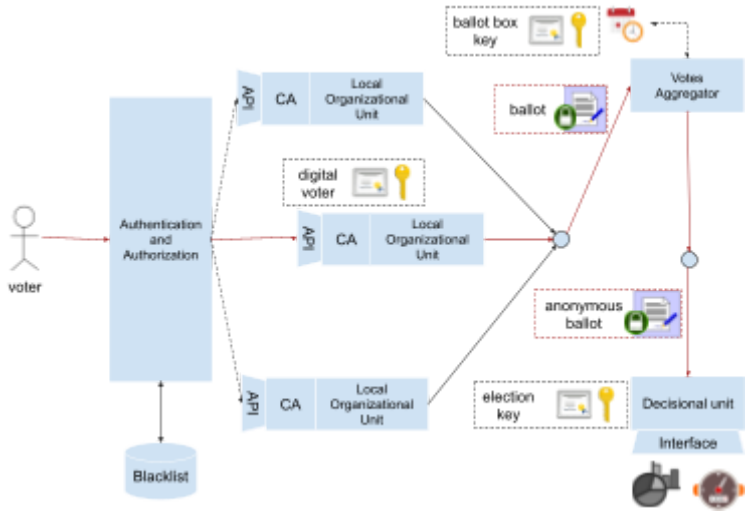

Figura 10 - Platforma de votare electronica - arhitectura concept [92]

Atunci când se definește arhitectura unui sistem electoral [92], este obligatoriu să se integreze modele de securitate în structura acestuia, încă de la început. Principala provocare atunci când se implementează un sistem de vot electronic la nivel național, pentru alegerile statale sau comunitare, este asigurarea unui echilibru între anonimat și riscul de a introduce în sistem buletine de vot neavizate și rău intenționate. Deoarece aceste 2 concepte pot intra cumva în conflict unul cu celălalt, activitatea de divizare a viziunii arhitecturale în fluxuri de date și blocuri funcționale complet izolate devine o cerință de facto. Prin urmare, atunci când se proiectează un sistem de vot electronic, un aspect principal este segregarea sarcinilor, pe de o parte din perspectiva logicii structurale și de procesare și pe de altă parte din perspectiva grupului de gestiune vizat. O organizație de stat ar trebui să evalueze partea de autentificare și autorizare și să translateze un alegător activ într-o entitate digitală, o altă organizație de stat ar trebui să se ocupe de fluxul buletinelor de vot anonimizate, verificând și eliminând orice informații cu caracter personal strecurate și transferând buletinele către o entitate colectoare. În cele din urmă, procesul electoral ar trebui să fie încheiat de o organizație specializată care trebuie să numere voturile anonime și să publice rezultatele.

Pentru a asigura anonimatul, mecanismul de autentificare și autorizare trebuie să fie complet detașat de sistemul principal de vot (Figura 10), reducând astfel la minimum spectrul șanselor de a putea corela persoana cu identitatea sa digitală. Mecanismul de autentificare și autorizare utilizează mijloacele legale pentru a identifica o persoană care are dreptul de a vota în cadrul procesului electoral actual. În plus, pe baza coordonatelor utilizatorului, are și rolul de a aloca identitatea digitală a unui alegător unei unități organizatorice locale pentru optimizarea procesului electoral și creșterea trasabilității în cazul unor încercări malițioase de a genera buletine de vot multiple (problema tranzacționării duble - double-spending [96]). Această euristică trebuie, de asemenea, să monitorizeze acei utilizatori care au fost trecuți pe lista neagră de către sistemele juridice și trebuie să fie în contact cu unitățile organizaționale locale pentru a revoca acele certificate care sunt asociate unui utilizator,

recent trecut pe lista neagră. În consecință, revocarea trebuie să conducă la anularea voturile generate de acel utilizator.

Odată transformat într-o entitate digitală, un cetățean primește o cheie privată și un certificat semnat de organizația locală, care pot fi utilizate în procesul de criptare pentru a semna buletinele de vot digitale ca parte a participării la procesul electoral în curs.

Procesul de votare în sine necesită 2 etape de criptare și ar trebui să permită participanților să își schimbe opțiunea înainte de data scadentă a alegerilor. Buletinul de vot, în sine, va fi împachetat într-un "plic dublu" [97] pentru a securiza informațiile și a proteja identitatea alegătorului în timpul procesului electoral. În prima etapă de criptare, informațiile stocate în buletinul de vot vor fi criptate cu cheia publică alocată pentru campania electorală în curs. Odată securizat, buletinul de vot trebuie să fie livrat către o unitate colectoare, flux de date care declanșează un al doilea proces de criptare. În această a doua etapă, obiectul generat anterior este semnat cu cheia privată a utilizatorului și trimis către o unitate colectoare, informațiile fiind criptate cu cheia publică a colectorului. Din punct de vedere pragmatic, urna de vot are propria pereche de chei cu scopul de a ascunde și proteja buletinele de vot în context digital. Cu toate acestea, în cazul în care urna de vot este compromisă, opțiunile participanților nu vor fi vizibile, deoarece cheia privată a campaniei electorale va fi găzduită de o altă unitate care are sarcina de a calcula și prezenta oficial rezultatele votului.

Agregatorul sau urna de vot digitală va primi în permanență tranzacții care conțin opțiunile alegătorului, păstrate într-un plic criptografic. Entitatea va monitoriza în mod activ intervalul de timp stabilit pentru procesul de votare și va lua în considerare doar cea mai recentă tranzacție primită de la o anumită entitate emitentă. Prin urmare, în cazul în care un alegător va dori să își schimbe opțiunea înainte de data scadentă a alegerilor, va fi luată în considerare doar cea mai recentă opțiune. Odată ce intervalul de timp alocat se scurge, o rutină va examina din nou toate buletinele de vot primite de entitatea colectoare, va extrage înregistrarea criptată a unui vot și va elimina semnătura utilizatorului. În acest moment, orice legătură cu sursa unui vot va fi eliminată. Entitatea colectoare va transmite toate buletinele de vot criptate către o unitate de decizie care va calcula și publica rezultatele alegerilor. Orice tranzacții noi provenind de la unitățile locale nu vor fi luate în considerare și nu vor fi trimise mai departe către unitatea decizională, ceea ce face imposibilă manipularea sistemului după expirarea perioadei de timp a alegerilor și sigilarea urnelor de vot digitale. Un aspect esențial în definirea unui astfel de model cu 2 fluxuri principale de date (de la organizațiile locale la unitatea colectoare și de la unitatea colectoare la unitatea decizională) este acela de a izola canalele de comunicare dintre unitățile organizatorice locale și unitățile decizionale. Urmând această abordare, fluxul de informații va fi ascuns pentru acele componente critice ale sistemului și, prin urmare, nicio unitate nu va putea să ruleze algoritmi complecși și să facă calcule paralele pe baza modelelor comportamentale. Desigur, dacă toate componentele sistemului vor fi gestionate de o singură autoritate și dacă securitatea digitală nu va fi dublată de lege, sistemul nu va fi complet inviolabil. Cu toate acestea, prin distribuirea responsabilității către unități de administrare specializate, astfel de topologii segregate pot deveni oportunități în proiectarea unor platforme de vot electronic la nivel național.

Structura propusă urmează unele povești de succes și modele de proiectare absorbite din implementările existente în țări precum Estonia, Norvegia sau Elveția. Cu toate acestea, o provocare interesantă rămâne structura mecanismului de autentificare. În acest caz, o implementare trebuie să aducă într-o locație la distanță sau la domiciliul alegătorului securitatea și confidențialitatea oferite de un centru de votare specializat. O opțiune poate fi adoptarea unui card electoral specific sau a unei chei stocate într-o carte electronică de identitate, ascunsă în spatele unui cod PIN și citită cu un cititor de cipuri USB specializat. O altă opțiune poate fi o semnătură digitală obținută de la o organizație certificată, înainte de startul procesul de votare.

O a treia opțiune, și cea care a fost luată în considerare de mine atunci când am proiectat arhitectura conceptului, a fost ideea de a obține controlul deplin al camerei web a dispozitivului de pe care un alegător decide să exprime opțiunile de vot și de a folosi această capacitate pentru a colecta informații esențiale. Înainte de a accesa interfața de vot, unui utilizator i se cere să scaneze actul de identitate sau pașaportul, prin utilizarea camerei web a dispozitivului și să pozeze pentru câteva fotografii de profil [98]. O metodă similară este utilizată în prezent de companiile care oferă servicii bancare alternative [99] pentru a deschide noi conturi de debit în numele unei persoane. Odată obținute toate datele de intrare necesare, algoritmi complecși de inteligență artificială pot asimila aceste informații și pot autentifica sau respinge accesul unui utilizator, încercând să găsească similitudini între identitatea preluată din documentele oficiale și fotografiile realizate cu camera web.

Un alt aspect esențial atunci când se detașează mecanismul de autorizare și autentificare de platforma principală de vot electronic este ușurința de gestionare atunci când procesul electoral rulează un model hibrid. De asemenea, existența unui sistem centralizat conectat la o bază de date de evidență a populației și capabil să înregistreze persoanele care își exprimă votul prin oricare din mijloacele existente, reduce semnificativ riscul de vot dublu. Acest model trebuie, de asemenea, să fie susținut de o metodologie care să stabilească prioritățile în cazul unei încercări de a vota prin toate mijloacele disponibile (de exemplu: voturi exprimate prin servicii poștale vs. vot electronic care poate fi ușor anulat prin revocarea certificatului; votul pe hârtie trebuie să înlocuiască orice altă metodă).

Întrucât contextul global ne-a arătat miza digitalizării în timpul crizei Covid-19, producerea unor soluții rapid pentru probleme complexe ar putea fi un obiectiv important al prezentului. O mare oportunitate este reutilizarea activelor existente în soluții și scenarii în care acestea pot aduce o contribuție semnificativă, chiar dacă nu au fost concepute pentru a se încadra perfect în context.

Ținând cont de acest principiu, am adoptat Blockchain și Hyperledger Fabric pentru a utiliza rutinele din compoziție, bine testate și mature, capabile să acopere o parte din elementele constitutive critice propuse în perspectiva arhitecturală.

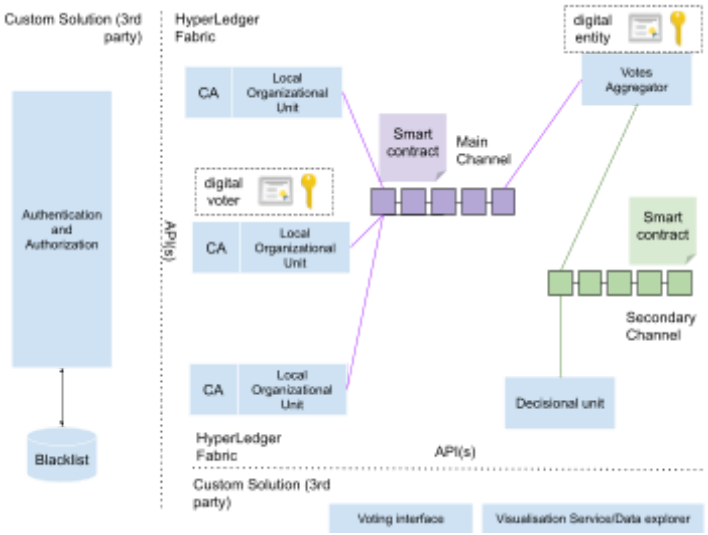

Figura 11 - Design peste Hyperledger Fabric [92]

Modelul adoptat de Hyperledger Fabric [36] pune accentul pe rolul organizării în structura unei topologii de comunicare. Fiecare macroentitate este reprezentată de o organizație în numele căreia, utilizatorii pot trimite informații în cadrul rețelei. O entitate organizațională își poate stabili propriile reguli de validare și politici de control de admitere și poate, de asemenea, să simuleze efectul pe care o tranzacție îl poate declanșa în cadrul rețelei. În plus, structura poate fi distribuită pe mai multe sisteme, nu neapărat colocalizate, care pot, de asemenea, să segmenteze responsabilitatea pentru

decizii specifice către entități de administrare specializate. Având o privire mai largă (Figura 11), pe lângă componenta participativă, o organizație are propria autoritate de certificare pentru a autoriza și valida utilizatorii abonați la ea. Pentru cazul meu de utilizare, am 3 tipuri de organizații, dar numai un singur tip expus alegătorilor pentru a-și exprima votul. Așa cum am descris anterior, există mai multe entități organizaționale locale care pot fi distribuite geografic, o unitate de colectare sau o urnă de vot digitală și o unitate decizională care trebuie să primească toate buletinele de vot și să poată calcula rezultatele alegerilor. Deoarece scopul modelului meu arhitectural este de a centraliza mecanismul de autentificare și autorizare, distribuția utilizatorilor și comunicarea cu Autoritățile de Certificare ale organizațiilor locale vor fi gestionate din exterior prin intermediul API-urilor Hyperledger. Astfel, un actor va trece mai întâi printr-un mecanism de autentificare gestionat de un terț, care va lua decizii pe baza coordonatelor primite de la acesta. Mai întâi, o euristică va poziționa utilizatorul în premisele unei organizații locale (de exemplu, pe baza adresei de domiciliu a utilizatorului). Odată luată decizia, aceeași entitate externă va declanșa autoritatea de certificare corespunzătoare pentru a genera o identitate digitală pentru cetățeanul autentificat. Aceeași entitate va monitoriza intrările de pe lista neagră și va solicita organizației corespondente să anuleze și să elimine drepturile de vot pentru o anumită identitate.

Odată autentificați, utilizatorii vor avea acces la o altă interfață externă, asociată cu hub-ul local. De aici, acesta va putea genera buletine de vot criptate și va putea declanșa tranzacții blockchain în numele organizației locale. Astfel, logica suplimentară care rămâne deasupra rutinelor existente ale Hyperledger Fabric, va fi capacitatea de a genera un buletin de vot secret criptat cu o cheie publică suplimentară, generată în prealabil pentru procesul electoral curent. Odată generat, acest bun purtător de valoare (sub forma unui mesaj criptat), va fi trimis prin blockchain către entitatea colectoare (urna de vot digitală) sub forma unei tranzacții blockchain. În acest moment, buletinul de vot va avea un dublu înveliș, întrucât se prezintă sub forma unui șir criptat, dublu-criptat ulteior de mecanismul de tranzacționare, cu cheia publică aparținând colectorului.

Euristica asociată fluxului de tranzacționare poate fi proiectată prin intermediul unor contracte inteligente personalizate complexe, întrucât implică mai multe constrângeri legate de aplicarea unui termen de viață pentru procesul electoral și de posibilitatea de a modifica votul inițial înainte de data limită. Astfel, contractul inteligent va trebui să interogheze un oracol pentru a monitoriza intervalul de timp (de exemplu, un server Network Time Protocol din exterior) și trebuie să anuleze activul purtător de valoare (buletinul de vot) ajunse la entitatea destinație în cazul în care mai multe tranzacții sunt generate de aceeași sursă. În plus, dacă intervalul de timp se scurge, contractele inteligente trebuie să oprească procesarea oricăror alte tranzacții, venite ulterior.

Un alt set de contracte inteligente ar putea fi, de asemenea, necesar în comunicarea dintre entitatea colectoare și unitatea de decizie. Odată ce intervalul de timp alocat alegerilor se încheie, acest set de rutine ar trebui să adune toate activele primite de unitatea colectoare, să elimine orice semnătură digitală și să le trimită, via tranzacții blockchain, în contul unității decizionale. Odată ajunsă acolo, unitatea decizională poate transmite informațiile către un alt serviciu terț sau poate rula un alt contract inteligent ca să decripteze buletinele de vot și să calculeze rezultatele.

Pentru a asigura o izolare completă a fluxurilor de comunicare stabilite între organizațiile locale și colector și între colector și unitatea decizională, Hyperledger oferă conceptul de canale. Prin urmare, o implementare care utilizează această caracteristică va avea două rețele blockchain diferite și două seturi de contracte inteligente care rezidă pe rețele logice blockchain anterior delimitate.

Deoarece Hyperledger Fabric este o tehnologie blockchain privată/prevăzut cu rutine de stabilire a permisiunilor, mecanismul de consens poate fi simplificat în funcție de nevoile logistice. În cazul meu, consensul ar trebui să fie stabilit de unitatea de primire, asemănător cu gestiunea lanțului de aprovizionare. Prin urmare, organizația de destinație ar trebui să valideze parametrii unei tranzacții și să propună blocuri.

Utilizarea blockchain poate crește, de asemenea, trasabilitatea în cazul unei anchete privind o fraudă, deoarece va fi mai ușor de urmărit fiecare etapă prin care a trecut un vot, chiar dacă informațiile înregistrate sunt secrete.

2.4. Prezentare generala a perspectivelor

În studiul de caz actual [92], am propus un model arhitectural, orientat pe tehnologia blockchain pentru un sistem național de vot. Proiectarea și deciziile arhitecturale au fost construite de jos în sus, pornind de la structura Hyperledger Fabric și încorporând modele din sistemele de vot existente. Noutatea propusă de mine rezultă din modelul de segregare a componentelor funcționale, din metoda de autentificare preluată din lumea soluțiilor financiare alternative și din simplitatea modelului ușor adaptabil la orice organigramă regională din structura unui stat. În plus, o implementare minimă a platformei de vot electronic a oferit o bună predicție a scalabilității. Mi-am bazat măsurătorile pe reglarea fină a parametrilor de configurare a Hyperledger Fabric și am identificat acele elemente de configurare care vor crește performanța generală a sistemului - blocuri cu multe tranzacții și baza de date GoLevelDB încorporată pentru gestionarea stării rețelei. Urmând un model distribuit și descentralizat, sistemul propus s-a dovedit a fi tolerant la erori și rezistent la formele de manipulare rău intenționate.

Euristica de tranzacționare aleasă, prin stabilirea a 3 faze ale procesului de vot, a stărilor de agregare și generalizare impuse și a tilizării metodologiei dublului plic, oferă o perspectivă granulară asupra stabilirii dimensiunii de anonimizare a datelor transmise.

3. Transportarea datelor

Un alt aspect esențial care trebuie prezentat în ceea ce privește confidențialitatea și securitatea datelor este modul în care datele sunt mutate peste rețea, menținând în același timp, în mod corespunzător, limitele proprietății și confidențialitatea acestora. Astfel de aspecte pot necesita o mai bună înțelegere a modului în care trebuie construită o topologie de comunicare pentru a acoperi în mod nativ astfel de cerințe. În plus, este, de asemenea, foarte important să se înțeleagă oportunitățile de scalabilitate și reziliența infrastructurii vizate în cazul unui atac asupra rețelei. Deși multe infrastructuri sunt capabile să gestioneze în mod corespunzător cerințele de confidențialitate și securitate, uneori acestea pot implica costuri considerabile de performanță sau un necesar consistent de resurse. În domeniul comunicațiilor, în special, trebuie să existe întotdeauna un compromis între politicile de securitate și confidențialitate și viteză, deoarece excesul uneia dintre ele afectează celelalte dimensiuni. Analiza completă a securității și a confidențialității în cazul evenimentelor unice sau corelate, capturate într-o rețea de comunicații, poate aduce un cost disproporționat de resurse și poate încetini fluxul de comunicație.

3.1. Întrebări de cercetare

- Cum se asigură autenticitatea[7] datelor și se modifică proprietatea datelor în comunicarea bizantină[8]?
- Care sunt oportunitățile și care sunt costurile pentru transportul datelor confidențiale și valoroase?
    - Cum se transferă activele digitale valoroase pe Internet?

[7] originea
[8] expus la o plajă largă de atacuri

3.2. Contribuţii

- Popa, Alin Bogdan, Ioan Mihail Stan, and Răzvan Rughiniş. "Instant payment and latent transactions on the Ethereum Blockchain." *2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2018.
- Stan, Ioan-Mihail, Ilie-Constantin Barac, and Daniel Rosner. "Architecting a scalable e-election system using Blockchain technologies." *2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2021.

3.3. Fundament tehnic

Pe lângă funcţia de stocare, Blockchain este o reţea punct-la-punct (peer-to-peer sau P2P) care asigură comunicarea între omologi prin intermediul tranzacţiilor. Astfel, fiecare tranzacţie (eveniment de comunicare) este validată de către comunitate şi stocată, în structuri de date numite blocuri, într-un registrul de evidenţă digital, pentru trasabilitate. Procesul de validare presupune ca destinatarii să îndeplinească condiţiile impuse de un contract inteligent pentru a procesa tranzacţia. Odată validată, o tranzacţie este efectuată numai atunci când s-a ajuns la un consens între participanţi, iar tranzacţia este listată într-un bloc nou generat, ataşat în continuare structurii blockchain (registru de evidenţă digital). În timpul efectuării unui proces de validare, timpii de procesare a tranzacţiilor pot creşte şi, prin urmare, comunicarea poate fi mai puţin eficientă. Cu toate acestea, în funcţie de implementare, diferite soluţii blockchain pot înregistra valori diferite ale timpului înregistraţi între două promovări de blocuri consecutive. În spectrul public, implementările pentru criptomonede sunt direct influenţate de mecanismul de consens. În Bitcoin, de exemplu, timpul dintre promovarea blocurilor consecutive este de aproximativ 10 minute [109], în timp ce în Ethereum, protocolul GHOST[110] suportă timpi de ataşare de blocuri mai rapizi, de aproximativ 10 până la 19 secunde. Pe lângă transferul de active digitale între omologi, o soluţie blockchain trebuie să se ocupe de schimbul de metadate/informaţii legate de tranzacţii sau blocuri, între toate entităţile participante. În plus, după cum au explicat V. Deshpande et al. [106] în enunţul problemei, fiecare entitate participantă trebuie să menţină, la nivel local, informaţii despre topologia de comunicare, completă sau parţială. Prin urmare, fiecare nod individual trebuie să se bazeze pe mecanismele de descoperire a participanţilor, pe mecanismele de filtrare a traficului de intrare, pe euristica de selecţie a vecinilor şi aşa mai departe. O parte din aceste acţiuni secundare au o influenţă directă asupra progresului general al comunicării. După cum subliniază V. Deshpande et. al., proprietăţile topologiei de comunicare sunt influenţate de tipul de blockchain, de mecanismul de consens, de numărul de conexiuni de intrare şi de ieşire şi de procesul de selecţie a vecinilor. În plus, lucrarea oferă măsurători care dovedesc modul în care implementările blockchains private pot gestiona mai puţini participanţi decât implementările blockchain publice, datorită modelului de consens mai puţin complex. În plus, autorii fac o distincţie între mecanismul de consens prin vot şi mecanismul de consens de tip loterie, acesta din urmă fiind capabil să atragă şi menţină mai mulţi participanţi în interiorul reţelei, decât celălalt model.

După cum definesc J. Spasovski et. al., în comparaţie cu implementările non-blockchain [108], soluţiile blockchain pot provoca întârzieri de până la 2-4 ori mai mari în timpii de răspuns şi au premisele să fie intolerante în caz de sarcină mare de trafic. Pentru a minimiza impactul mecanismului de consens, autorii efectuează analize pe implementări blockchain care utilizează algoritmul/protocolul proof-of-stake (dovadă a implicării cu active) de consens, mecanism care conservă puterea de procesare şi angajează entităţile participante să implice averea pentru stabilizarea reţelei. Deoarece averea lor este direct afectată de comportamentul manifestat în sistemul distribuit şi descentralizat, implementarea unui astfel de algoritm facil de consens generează un consum mai mic de energie şi provoacă mai puţină congestie. Cu toate acestea, după cum subliniază autorii, soluţiile

blockchain se extind liniar şi, cu suficiente noduri care se alătură reţelei, o astfel de implementare poate înregistra în continuare o telemetrie bună: debit ridicat şi timpi de răspuns mici. Având în vedere valorile intrinseci ale blockchain în ceea ce priveşte imutabilitatea, acesta poate fi în continuare o potrivire perfectă în diverse scenarii, inclusiv în industrii critice axate pe tranzacţii securizate.

Blockchain a devenit o opţiune potrivită pentru a asigura o platformă de comunicare adecvată şi sigură peste "internetului lucrurilor" (internet of things), care să susţină datele voluminoase, în special în spectrul privat. După cum J. Zhang et al. prezintă în lucrarea [107], blockchain implementează modelul arborelui de comunicare concurentă şi, de asemenea, este capabil să distribuie funcţia de stocare către mai multe noduri agregatoare, impunând, de asemenea, o securitate şi o confidenţialitate corespunzătoare asupra datelor transmise. În reţelele blockchain private, dotate cu sistem de permisiuni, utilizatorii pot adera la reţelele organizaţiilor pentru a colecta informaţii de la senzori şi dispozitive IoT [111][112]. Astfel, acestea pot fi utilizate cu succes în implementarea oraşelor inteligente pentru a colecta telemetrie relevantă şi pentru a o împărtăşi în siguranţă informaţia către public. În plus, poate fi utilizate pentru a comercializa energie în microreţele, susţinând inovaţia din industria energetică.

Reţelele Blockchain publice sunt cea mai cunoscute implementări blockchain de către mase şi sunt utilizate în principal în criptomonede. Odată cu apariţia Ethereum şi a limbajului de programare a contractelor inteligente [113] - Solidity -, s-a născut o nouă eră a consumului de servicii online. Astfel, Ethereum, prin intermediul DApps [114], este acum capabil să furnizeze putere de procesare la cerere, fără a fi nevoie ca un furnizor de servicii să deţină infrastructura de bază. Cu toate acestea, de-a lungul timpului, soluţiile blockchain nu au fost pe deplin protejate împotriva atacurilor asupra reţelei, atacurilor asupra contractelor inteligente sau a atacurilor asupra euristicii de consens [115][116][117] Uneori, astfel de atacuri au dus la coruperea efectivă a registrului descentralizat. Una dintre problemele clasice este legată de oportunităţile de dublă tranzacţionare, atunci când, în cazul criptomonedelor, un participant păcăleşte reţeaua şi reuşeşte să cheltuiască acelaşi jeton digital, purtător de valoare în două tranzacţii distincte. Două dintre aceste probleme de securitate a tranzacţiilor [115] sunt reprezentate de Finney Attack şi Race Attack [117], ambele utilizând un defect logic timpuriu în procesarea tranzacţiilor. În actualizările recente, astfel de probleme nu mai sunt posibile, deoarece acum procesarea tranzacţiilor este limitată de o euristică de confirmare. Cea mai problematică situaţie, de asemenea legată de securitatea tranzacţiilor, rămâne problema 51% [117]. Acest lucru se întâmplă dacă un participant sau o sub-comunitate de participanţi controlează mai mult de jumătate din reţea. Aceste carteluri cibernetice, în teorie, pot modifica datele stocate în blockchain, deoarece capitalizează mai multe rate hash (rată de generare a sumelor de control a datelor) decât ceilalţi mineri.

Spectrul atacurilor de reţea este, de asemenea, cuprinzător, fiind cel mai relevant în contextul comunicării. Categoriile de atacuri sunt, de asemenea, moştenite de la vechile soluţii P2P, utilizate în alte domenii. O problemă relevantă este atacul Sybil, în care un utilizator rău intenţionat poate crea şi se poate ascunde în spatele mai multor identităţi. O altă problemă relevantă privind securitatea reţelei este atacul Eclipse [115][116], un atac ţintit cu scopul de a deconecta un nod de la blockchain şi de a-l face să comunice şi să facă schimb de informaţii doar cu noduri rău intenţionate. Astfel de probleme pot fi rezolvate dacă nodul este capabil să identifice vecinii legitimi şi să respingă orice alt trafic de intrare. În plus, ca în orice configuraţie expusă public, toate nodurile participante pot fi supuse atacurilor de suprasolicitare (distributed denial of service), lansat în mod distribuit.

Este posibil ca o parte din problemele de securitate prezentate anterior să nu fie relevante în implementări blockchain private/ dotate cu mecanism de permisiune, deoarece, în majoritatea cazurilor, acestea sunt reţele închise în care participanţii sunt certificaţi în prealabil, înainte de a se alătura comunităţii. Reţelele Blockchain publice nu sunt dotate cu mecanisme de stabilire a permisiunilor asupra datelor, ceea ce înseamnă că oricine se poate alătura. Totodată, orice schimb de

informații trebuie validat de mineri, fapt ce poate implica o putere de procesare sporită pentru a proteja rețeaua. Toate informațiile sunt publice și oricine poate citi și scrie în registrul distribuit și descentralizat. Rețelele blockchain private, dotate cu mecanisme de permisiuni se potrivesc mai bine unor cazuri particulare de utilizare, în care preocuparea nu este legată de o validare complexă a tranzacțiilor, ci de trasabilitatea mutării activelor digitale critice în interiorul unor comunități închise de participanți. Costul de funcționare a mecanismelor de consens, cum ar fi Proof-of-Work, capabile să acopere întregul context bizantin, poate să nu fie fezabil, prin urmare sunt necesare mecanisme de consens mai facile din punct de vedere computațional. Algoritmi similari cu RAFT[118] care implementează paradigma selectării unui lider sau algoritmi de selecție a unui promotor de blocuri, pe bază de loterie, oferă suficientă certitudine în ceea ce privește protecția datelor stocate în rețelele blockchain private.

3.4. Studiu de caz - Plata instantanee în tranzacțiile cripto-financiare prin intermediul Ethereum Blockchain

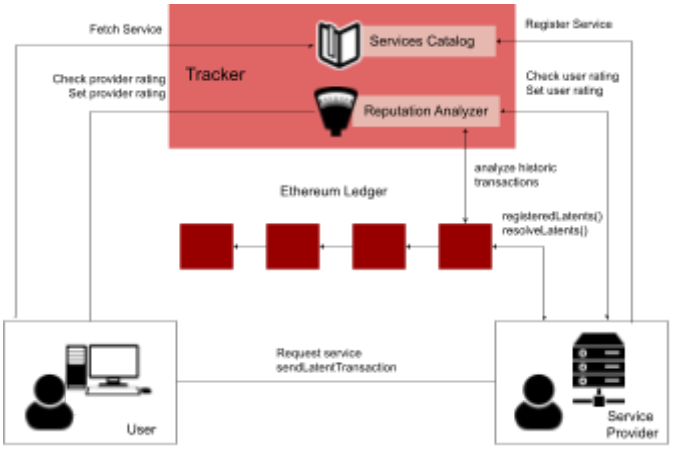

Figura 12 - Sistem de tranzacții latente - arhitectură concept

Implementări precum Ethereum [120], care oferă mecanisme de extindere a funcționalității de bază a rețelei prin contracte inteligente Turing-Complete și DApps, au deschis o nouă piață pentru servicii digitale, închiriind putere de procesare în schimbul remunerării cu criptomonede. Prin urmare, prin intermediul DApps, în teorie, se poate accesa o mare varietate de servicii dezvoltate ca un strat suplimentar peste rețeaua publică. Chiar dacă Ethereum înregistrează timpi de promovare de blocuri destul de mici, pentru serviciile în timp real încă nu oferă perspective bune. Prin urmare, o metodă de a depăși problemele legate de viteză este dezvoltarea unei infrastructuri complementare într-un mix hibrid. Prin urmare, ideea inovatoare la care am contribuit poartă numele de tranzacții latente [119] și implementează un serviciu existent în sistemele bancare și de asigurări, pe rețeaua Ethereum. O tranzacție latentă este un acord în afara rețelei blockchain care oferă garanții atât consumatorului, cât și producătorului/facilitatorului ca, în schimbul unor servicii în timp real, o tranzacție de remunerare va fi procesată la un moment ulterior. Cele două garanții privind consistența acordului sunt asigurate de standardul ERC20 [121] (pentru dezvoltarea contractelor inteligente) care oferă mijloacele de a controla soldul partenerilor odată ce contractul este activat, și de un sistem alternativ de evaluare pentru ambele părți implicate, bazat pe experiența anterioară (Figura 12). În plus, pe lângă noile tranzacții latente, configurația hibridă propune, de asemenea, o platformă care afișează un catalog de servicii, gata să primească noi furnizori de servicii în timp real (ca, de exemplu, administratori de servere de jocuri sau furnizori de servicii de streaming video).

În conformitate cu standardele industriei în ceea ce privește serviciile în timp real, au fost stabilite cinci criterii de acceptanță distincte pentru sistemul de remunerare:
Q1 - Soluția trebuie să fie capabilă să expună o varietate infinită de servicii sau operațiuni
Q2 - Încrederea într-un furnizor trebuie să se bazeze pe randament (rating) și analiză
Q3 - Plata serviciilor trebuie să fie adaptabilă pe baza cererii sau a efortului de calcul.
Q4 - Sistemul trebuie să permită plata/remunerarea instantanee.
Q5 - Sistemul trebuie să accepte cereri și să furnizeze servicii chiar dacă părțile interesate sunt deconectate de la reteaua blockchain publică, în timpul schimbului de date.

În concluzie, studiul de caz prezintă o soluție care vizează rezolvarea problemei latenței tranzacțiilor în Ethereum Blockchain și în rețele similare. Prin urmare, a fost prezentat un concept nou numit tranzacții latente care externalizează schimburile de lichidități către o platformă din exteriorul rețelei blockchain, în timp ce remunerarea efectivă stabilită poate fi înregistrată în rețeaua Ethereum la un moment ulterior. Configurația suportă prețuri adaptive și este concepută pentru furnizorii de servicii în timp real. Soluția a fost dezvoltată în conformitate cu standardul ERC 20 pentru contracte inteligente. Totodată, pentru a depăși posibilele probleme de tip tranzacționare dublă, datorate externalizării, sistemul propus implementează, de asemenea, un serviciu de rating, pentru a evalua nivelul de încredere ce poate fi acordat entităților participante.

3.5. Discuție

În ceea ce privește studiul de caz prezentat în secțiunea precedentă - Anonimitatea datelor, rețelele blockchain cu sistem de permisiuni reprezintă o subcategorie a implementărilor blockchain care sunt concepute pentru a funcționa în contexte private și care oferă rețelei un mecanism suplimentar pentru gestiunea accesului. Prin urmare, fiecare participant, care se alătură rețelei, trebuie să obțină o identitate digitală de la administratorul infrastructurii blockchain, livrată prin certificate de tip x509. Fiecare certificat încorporează un set de parametri care definesc permisiunile de care dispune un participant. Astfel de modele de securitate oferă o confidențialitate mai bună decât implementările publice, deoarece, fiecare membru are acces granular la informațiile stocate în blockchain și are, de asemenea, un rol dedicat și prestabilit în cadrul rețelei. Cu toate acestea, deoarece toți participanții sunt cunoscuți și prevalidati, astfel de soluții nu respectă întru totul principiul anonimității. În plus, consensul și validarea datelor sunt gestionate de un set prestabilit de participanți [122]. Astfel, în cazuri foarte specifice, acest lucru poate aduce probleme, deoarece părțile implicate și desemnate pot modifica mecanismul de consens, politica de aderare sau ciclul de viață al contractelor inteligente[123] în timp real. În majoritatea implementărilor, în sectoarele financiare și nu numai, astfel de rețele blockchain sunt întreținute de consorții de organizații, care fac schimb de bunuri digitale și informații valoroase, validate de un set de membrii autorizați și certificați și nu de entități miner, ca în spectrul public. După cum menționează M. Cash et al. [124], nu toți participanții au dreptul de a citi sau de a scrie în blockchain. Prin caracterul lui privat, nu înseamnă că un blockchain nu poate fi expus în exterior, ci înseamnă că nu toți participanții au posibilitatea să manipuleze datele din registrul de evidență digital. Alte oportunități asociate cu rețelele blockchain prevăzute cu sistem de permisiuni este semi-descentralizare, Astfel nu sunt necesare comunități largi de participanți pentru a securiza rețeaua. Acest lucru o face să funcționeze mai rapid și impune implementarea de mecanisme de consens mai ușoare. În plus, prin natura sa de a putea fi modelată în timpul rulării, în funcție de obiectivele entităților sau organizațiilor participante, implementările în spectrul privat sunt foarte ușor de personalizat și oferă mecanisme de interoperabilitate. Cu toate acestea, chiar dacă, în ceea ce privește comunicarea, implică un model de încredere și o gestionare a accesului adecvată pentru acoperirea majorității preocupărilor legate de confidențialitate, astfel de soluții au propriile capcane. Astfel, prin limitarea descentralizării, se constituie oportunități de

corupere a rețelei, deoarece securitatea este gestionată de un număr mai mic de entități. Dacă sunt compromiși, aceştia pot afecta cu uşurință integritatea datelor. Cu toate acestea, astfel de situații sunt mai puțin probabile, deoarece majoritatea implementărilor nu tranzitează spre centralizare, ci urmează o descentralizare incrementală.

Soluțiile blockchain sunt potrivite pentru a furniza topologia de comunicare pentru tranzacționarea informațiilor valoroase, schimbate în industrii critice sau necritice, în special atunci când se utilizează implementări cu mecanisme de permisiuni. Cu toate acestea, o piedică este faptul că poate implica o putere de calcul și o capacitate de stocare consistente pentru un singur serviciu care asigură comunicarea și trasabilitatea. Astfel, în multe cazuri, spectrul limitat de servicii furnizate în contrast cu infrastructura implicată ar putea să nu fie fezabil, deoarece costul de întreținere a unei astfel de rețele ar putea să nu fie avantajos. Chiar dacă prin intermediul contractelor inteligente complexe se poate extinde spectrul de servicii consumabile, acestea sunt totuși limitate de procesarea lentă, de dezvoltarea și întreținerea constrânsă a codului peste timp și, de asemenea, de imuabilitatea intrinsecă, care nu permite corectarea la cald, a contractelor existente[125]. Prin urmare, pentru aceleași resurse implicate, se poate implementa paradigma cloud, care poate extinde semnificativ portofoliul de servicii și poate oferi, de asemenea, diverse servicii de comunicare care pot fi utilizate. O altă problemă referitoare la Blockchain este legată de conformitatea cu GDPR [126], unde dreptul de a fi uitat nu poate fi implementat cu uşurință atâta timp cât datele istorice trebuie păstrate în interior ca parte a atributului de imuabilitate. În contextul legal, dovada de ardere/ștergere a cheilor criptografice care asigură accesul la datele private, nu este pe deplin conformă cu reglementările, deoarece datele criptate sunt în continuare disponibile și partajate între nodurile participante. Cu toate acestea, în configurațiile hibride, datele reale pot fi găzduite în surse de date externe, aliniate cu reglementările, în timp ce o amprentă digitală a acestora poate fi menținută în rețelele blockchain publice. După cum sugerează N.B. Truong et. al., în rețelele blockchain prevăzute cu sisteme de permisiuni, multe probleme de conformitate pot fi rezolvate prin stabilirea corespunzătoare a rolurilor specifice GDPR (procesatori de date, controlori de date, subiecți ai datelor etc.) și prin dezvoltarea unui sistem de gestiune extern al accesului adițional celui asigurat de sistemul blockchain, incluzând o intermediere a oricărui API expus.

3.6. Prezentare generala a perspectivelor

Pentru a rezuma informațiile prezentate, blockchain este un excelent punct de plecare pentru stabilirea unei topologii de comunicare care asigură în mod nativ nevoia de securitate și confidențialitate a datelor transportate. Cu toate acestea, expunerea în spațiul public aduce un nou set de metode de atac, ușor diferit de modelele clasice de atac peste rețea. Chiar și în cazul expunerii private, riscurile de manipulare a datelor sunt încă prezente. Cu toate acestea, tehnologia oferă o protecție suficientă din start, reducând efortul de a dezvolta metode suplimentare pentru a completa deficiențele. Pentru migrarea controlului securității și confidențialității asupra datelor transportate, în premisele serviciului furnizat, paradigma cloud poate fi un facilitator mai bun pentru funcția de comunicare.

Rețelele publice sunt un motor excelent pentru transferul securizat de active digitale peste Internet și pentru stabilirea proprietății acestora. Cu toate acestea, costul de calcul, costul financiar și diversitatea tipurilor de atacuri în rețea, în comparație cu cele obișnuite, îl fac adesea nefezabil.

Rețelele blockchain private rezolvă o parte importantă a problemelor identificate în spectrul public. Acestea reduc costurile de calcul prin utilizarea unor protocoale de consens ușoare și se adresează comunităților închise. Cu toate acestea, tinde să încalce granița anonimatului datelor datorită caracteristicilor rețelei (puțini participanți, comunități închise) apartenența făcându-se prin mecanisme de certificare. De asemenea, întreținerea poate fi ușor complexă în contrast cu ansamblul

de funcții pe care le oferă peste infrastructura implicată. Astfel, cloud-ul devine relevant din prisma elasticității și versatilității.

4. Distribuția și orchestrarea datelor

Un alt subiect dezvoltat în jurul confidențialității datelor este orchestrarea și distribuția datelor, o problematică ce se referă (și pune accentul) pe aspecte precum localitatea și mobilitatea datelor și modul în care orchestrarea inteligentă poate comisiona și scoate din uz date și servicii de date, cu scopul asigurării disponibilității, securității și confidențialității. Astfel, după cum s-a menționat anterior, este important să se dispună de mijloace pentru a constrânge procesul de poziționare a informațiilor sensibile, modul în care sunt procesate informațiile și unde și cum se poate asigura distribuția pe mai multe locații în sistemele distribuite, prin intermediul unei orchestrații inteligente. Prin urmare, explorez posibilitatea de a îmbina diferite sisteme de guvernanță a datelor, cum ar fi rutinele de control cloud sau mecanismul de programare și gestionare din sistemele de calcul de înaltă performanță și a celor de tip Grid Computing (calcul peste rețea), cu scopul de a consolida, într-o soluție unificată, cele mai bune elemente din toate paradigmele. În plus, ca parte a mobilității datelor, prezentată pe scurt prin prisma transmiterii și expunerii datelor, mă concentrez pe validarea consistenței motoarelor de containerizare din industrie în furnizarea de date prin intermediul containerelor. Prin urmare, pe de o parte, validez cum procesatorii de date sunt încărcați în containere, entități capabile să ruleze pe sisteme eterogene, în apropierea datelor. Pe de altă parte, validez modul în care containerele pot păstra confidențialitatea folosind taxonomia clasică de izolare versus o taxonomie mai slabă. Un container este o metodă de virtualizare ușoară asigurată de mecanisme specifice rezidente în nucleul sistemului gazdă. Într-o definiție mai pragmatică, un container poate fi văzut ca o multiplicare a spațiului utilizatorului. Din punct de vedere al securității, evadarea dintr-un container implică un risc ridicat de corupere a întregului sistem gazdă, deoarece partea cea mai critică a acestuia este partajată între alte containere și spațiul utilizatorului.

4.1. Întrebări de cercetare

- Cum pot orchestra atât datele, cât și procesatorii de date pe sisteme distribuite mari?
- Cum poate fi asigurată localitatea[9] datelor în infrastructurile critice?

4.2. Contribuții

- Stan, Ioan-Mihail, Daniel Rosner, and Ştefan-Dan Ciocîrlan. "Enforce a global security policy for user access to clustered container systems via user namespace sharing." *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2020.
- Stan, Ioan-Mihail, Ştefan-Dan Ciocîrlan and Răzvan Rughiniș. “UNDERSTANDING THE OPPORTUNITIES OF APPLYING KUBERNETES SCHEDULING CAPABILITIES IN HIGH PERFORMANCE COMPUTING” *Scientific bulletin. Series C: electrical engineering and computer science*

[9] mutarea procesării în apropierea datelor

4.3. Studiu de caz - Unificarea spațiului de nume de utilizator pentru o mai bună securitate și confidențialitate

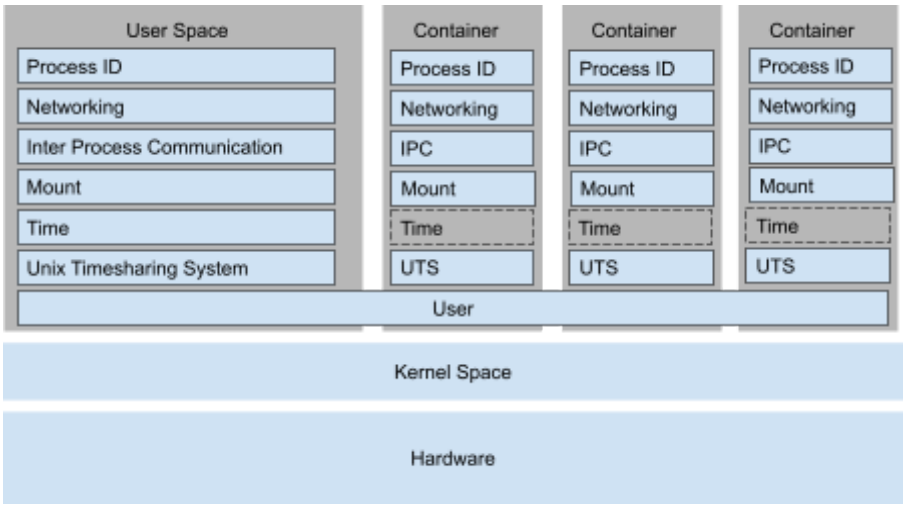

Figura 13- Taxonomia de izolare a containerelor [135]

Sistemele de operare moderne, utilizate pe scară largă, sunt în cea mai mare parte monolitice, ceea ce înseamnă că între hardware (fizic sau emulat) există un strat de abstractizare numit spațiu Kernel. Acest strat include toate driverele, planificatoarele și rutinele care interacționează în mod optim cu hardware-ul. Spațiul Kernel oferă servicii stratului superior prin intermediul apelurilor de sistem și impune un model de inel de protecție pentru componentele gestionate. Stratul superior și cel care servește drept interfață între software-ul utilizatorului și sistemul de operare se numește spațiu utilizator.

Domeniul de aplicare sau contextul unui spațiu utilizator este asigurat de 2 caracteristici/servicii importante implementate în nucleu (Linux Kernel): cgroups și kernel namespaces[136]. Acestea pot limita domeniul de aplicare al unui spațiu utilizator și propune o taxonomie de izolare[136] (Figura 13 - taxonomia de izolare a containerelor cu spațiul de nume utilizator partajat). Atunci când vorbim despre containere aflate în execuție, pot fi definite ca o instanță separată a unui spațiu utilizator - Figura 13. În acest sens, prin aplicarea aceleiași taxonomii de izolare, noua instanță a unui spațiu utilizator va avea propriile obiecte și mecanism de indexare și va moștenii în mare parte același rol ca și spațiul utilizator principal, nativ.

Prin mutarea punctului de observației către spațiul de nume al utilizatorului, acesta acoperă independent toate aspectele gestionării utilizatorilor, inclusiv sistemul de indexare a utilizatorilor (UID). Prin urmare, un utilizator cu UID 1000 în spațiul utilizatorului de sistem este diferit de un utilizator cu UID 1000 definit într-un container. Astfel, o politică de utilizator aplicată la nivelul sistemului de operare poate necesita adoptarea și reimplementarea la nivelul containerului.

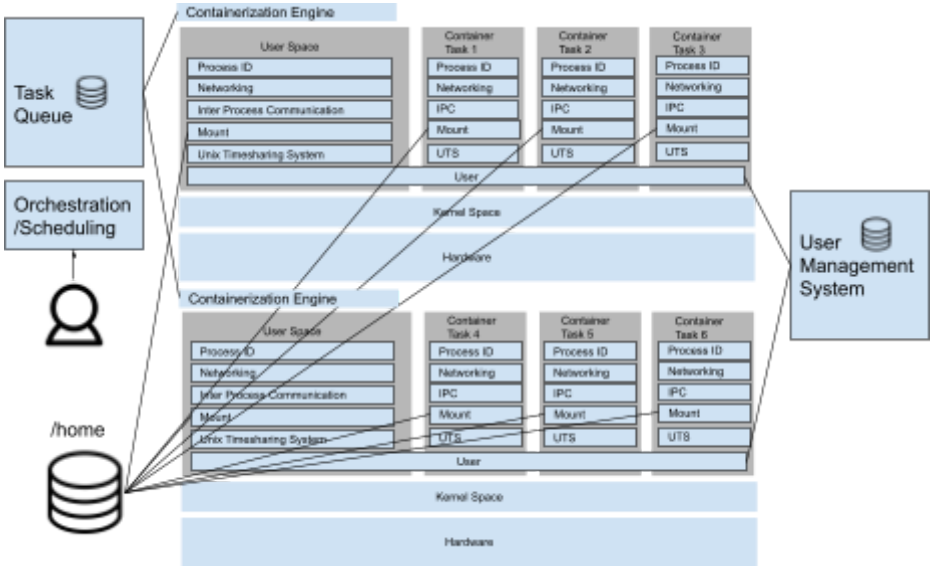

Figura 14- Arhitectura concept pentru partajarea spatiului de nume utilizator [135]

Gestionarea accesului și a utilizatorilor este una dintre cele mai importante primitive atunci când se proiectează un sistem securizat cu mai mulți chiriași și mai multe noduri - Figura 14. Uneori, aceasta poate implica și politici speciale de calitate a serviciilor (QoS) pentru granularitatea accesului, care pot fi implementate în mod distribuit sau centralizat. Odată cu avansarea tehnologiilor de containerizare, paradigma de securitate a trebuit să fie extinsă, deoarece, prin pornirea unui container conceptual/brut, in teorie, se poate impersona ansamblul nativ al spațiului utilizatorului. Prin urmare, un utilizator de container ar putea fi capabil să ocolească unele dintre constrângerile care au fost implementate la nivel de sistem, deoarece acestea nu se aplică în contextul altor spații de nume de utilizator, izolate. În plus, în cazul în care un proces se bazează pe gestionarea accesului la sistem pentru a lua anumite decizii, prin rularea unei instanțe a procesului respectiv într-un container, pe un nod fizic restricționat, se poate falsifica cu ușurință mecanismul de indexare a utilizatorilor și se poate masca o intenție rău intenționată.

Din punct de vedere conceptual, acest aspect de securitate poate avea două rezolvări simple, în raport cu capacitățile pe care le pot oferi motoarele moderne de containerizare:

- limitarea accesului la o listă de persoane esențiale pe nodurile de procesare și delegarea consolidării contextului și politicilor utilizatorului către un orchestrator extern
- partajarea spațiului de nume de utilizator al sistemului gazdă cu enclavele virtuale (containere)

Pentru prima soluție, un motor/orchestrator inteligent poate aplica un mecanism de re-mapare a utilizatorilor, ce va construi un canal de legătură între utilizatorii definiți la nivelul containerului și un subset predefinit de ID-uri de utilizator neprivilegiat (UID) definite la nivelului contextului utilizator din sistem. Prin urmare, toate încercările de accesare a resurselor sistemului vor fi redirecționate prin intermediul spațiului de utilizator al sistemului gazdă și executate fără privilegii suplimentare. O altă metodă de a depăși această provocare este manipularea corespunzătoare a contextului implicit al utilizatorului din container, la pornire. Prin urmare, chiar dacă imaginea originală a containerului aplică un context de utilizator specific, acesta va fi modificat la lansarea containerului. Această abordare este implementată în prezent de unele orchestratoare de containere (de exemplu, OpenShift OKD, Kubernetes). Cu toate acestea, uneori, acest model poate aduce unele probleme de compatibilitate. Cea mai frecventă problemă este atunci când procesele/aplicațiile au fost proiectate pentru a rula sub contul root în timp ce politica de securitate aplicată forțează aplicația să

ruleze ca un cont non-root. Aplicația nu va putea rula, iar containerele vor ajunge într-o stare de eroare.

În cazul celei din urmă opțiuni, prin simpla atașare a spațiului de nume al utilizatorului de sistem la toate containerele (Figurile 13 și 14), toate procesele vor moșteni același set de permisiuni și reglementări de la utilizatorul care a solicitat executarea containerului. În ciuda modelului ideal prezentat anterior, într-o abordare pragmatică, ar putea fi necesare și alte spații de nume pentru a se contopi parțial în sistem. Este posibil ca acestea să trebuiască să furnizeze servicii sau obiecte suplimentare pentru a sprijini aplicarea unei anumite politici sau setări (de exemplu, pentru gestionarea utilizatorilor, fișierele /etc/passwd și /etc/shadow trebuie să fie montate în mod obligatoriu în containere, cu acces numai pentru citire).

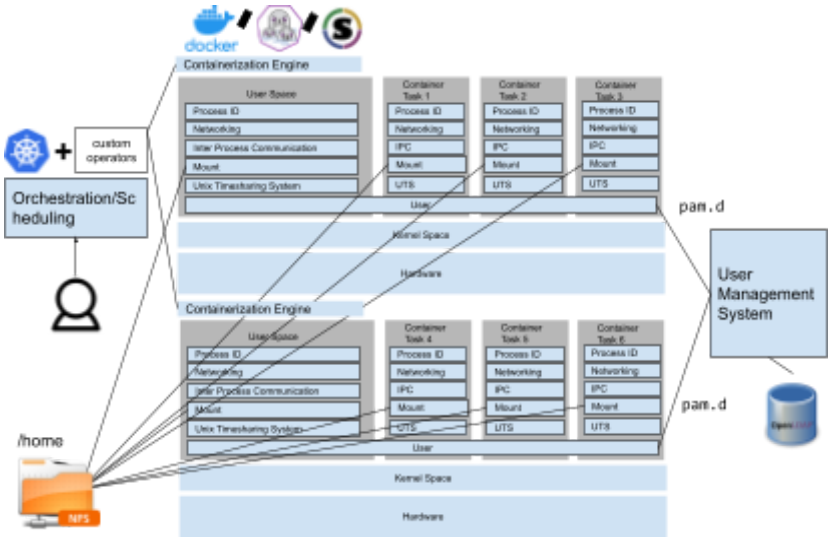

Figura 15 – Design arhitectură concept [135]

Pentru a răspunde nevoilor de proiectare (Figura 15) în cadrul arhitecturii propuse, voi analiza diferite motoare de containerizare (Docker, Podman și Singularity) și voi dezvălui modul în care structura lor poate încorpora uneori un risc de ocolire a limitelor de acces prin expunerea oportunităților de escaladare a privilegiilor. Voi testa modul în care diferite motoare de containerizare pot fi manipulate pentru a rula containere cu un context de utilizator unificat și voi identifica care sunt acele servicii și artefacte care trebuie să fie partajate între spațiul utilizatorului și container. Arhitectura propusă urmărește să ofere infrastructurilor critice mijloacele de a asigura un context unificat de politici de utilizator pentru a menține în mod corespunzător bunurile private și procesarea acestora în contexte izolate și proprietare. Din lista propusă, Singularity acționează mai bine, deoarece taxonomia sa nativă de izolare propune o izolare pe 1 strat prin intermediul spațiului de nume mount. Docker, din cauza arhitecturii sale client-server, poate prezenta un risc, deoarece poate fuziona cu contul root al sistemului, ca parte a unificării spațiului de nume de utilizator. Daemonul Docker rulează ca root, în timp ce clientul poate trimite cereri de la utilizatori fără privilegii. Podman rulează direct din contextul utilizatorului, fiind conceput ca o soluție exclusiv client. Prin urmare, în cazul rulării containerelor cu un spațiu de nume de utilizator unificat, maparea va reveni la utilizatorul de sistem, care a solicitat crearea containerului.

În acest studiu de caz [135], am prezentat diferite motoare de containerizare și constrângerile acestora în ceea ce privește ușurința de fuzionare a spațiului de nume al utilizatorului containerului cu spațiul de nume al utilizatorului din sistemul gazdă. Am prezentat faptul că, în contextul unor politici QoS foarte complexe, adaptate pentru fiecare individ în parte, existența unui spațiu de nume de utilizator comun, partajat între toate entitățile virtuale și sistemul de operare, poate crește securitatea

întregului ansamblu. Mai mult, extinzând acest scenariu la topologii cu mai multe noduri şi mai mulţi chiriaşi, am identificat o configuraţie comună de clustere în care o astfel de abordare poate fi esenţială - clusterele de calcul de înaltă performanţă.

Cel mai bun motor de containerizare în ceea ce priveşte studiul meu de caz este Singularity. Acest motor oferă în mod nativ funcţia de partajare a spaţiului de nume al utilizatorului. Soluţia este, de asemenea, compatibilă cu artefactele dezvoltate pentru alte motoare de containerizare (de exemplu, imaginile Docker). Pentru Docker şi Podman, rezultate similare (fuziunea spaţiului de nume de utilizator) pot fi obţinute printr-o secvenţă de paşi manuali sau prin intermediul unui Orchestrator care poate replica şi propaga politicile de utilizator pe fiecare container, în timpul lansării în context de rulare.

În cele din urmă, în studiul de caz curen [135]t, propun o arhitectură conceptuală şi un design pentru un cluster multi-nod, multi-chiriaş, care poate susţine unificarea spaţiului de nume de utilizator. Am folosit OpenLDAP pentru a externaliza gestionarea utilizatorilor cu posixAccounts, am utilizat un sistem de fişiere de reţea pentru a partaja orice artefact critic al utilizatorului între toate containere şi am folosit Kubernetes pentru a orchestra aplicarea contextului de utilizator adecvat pe fiecare container.

4.4. Studiu de caz - Adoptarea Kubernetes în sisteme de calcul de înaltă performanţă

Calculul de înaltă performanţă este unul dintre cele mai prolifice concepte din domeniul tehnologiei informaţiei şi unul dintre motoarele inovării. Încă din primele etape ale pandemiei Covid-19, companiile şi instituţiile au reunit un număr incredibil de resurse pentru a efectua studii asupra noului virus. Consorţiul HPC Covid19 a împărţit în mod gratuit aproximativ 6,4 milioane de nuclee CPU, 603 Petaflops şi 4,9 mii de GPU pentru realizarea proiectelor legate de Covid-19. Fiind încă o infrastructură rigidă, HPC a început să adopte containerul ca unitate de procesare, inspirat de beneficiile pe care acestea le-au adus în cloud.

Un alt inovator important pe piaţa sistemelor distribuite este Kubernetes, un orchestrator de containere, care facilitează implementarea de cloud pe sistemele private, cu o rată de adopţie consistentă la nivel mondial. După cum a anunţat CNCF (Linux Foundation), în prezent există aproximativ 5,6 milioane de dezvoltatori care au adoptat Kubernetes la nivel global. Cu un plan de control bine pus la punct şi un ecosistem construit în jurul tehnologiei, Kubernetes surclasează alţi competitori şi devine un standard al industriei.

Întrucât ambele platforme bazate pe sisteme distribuite îşi concentrează eforturile asupra containerelor, studiul de caz actual propune o clasificare a implementărilor hibride HPC-Kubernetes şi a modului în care au fuzionat parte din rutinele de control importante ale celor două sisteme. Această taxonomie îşi propune să analizeze implementările găsite în literatura de specialitate, pentru a le grupa.

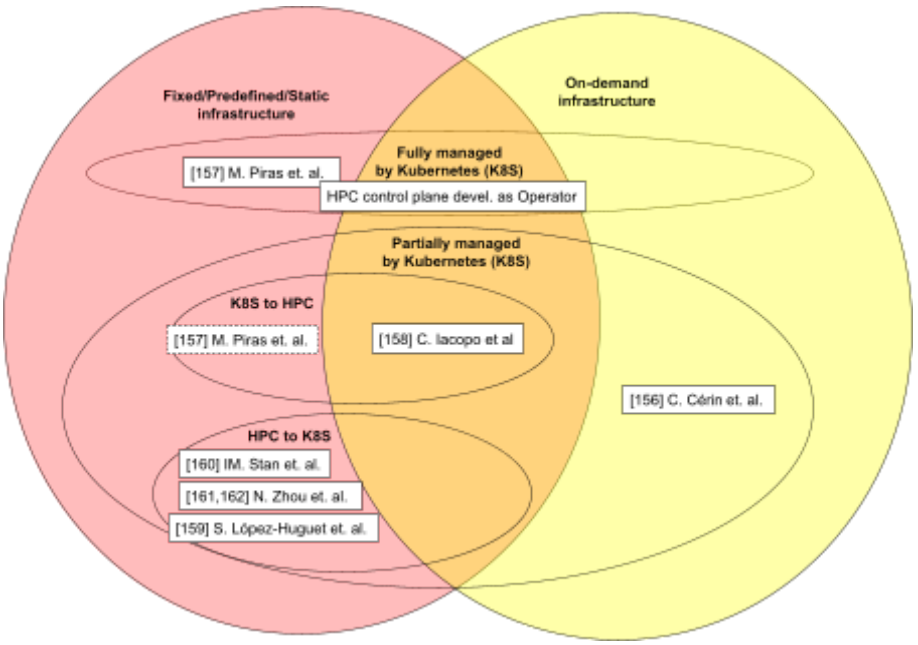

Figura 16 - Model de clasificare pentru sistemele hibride de tip HPC-Kubernetes

Metodologia de clasificare cuprinde o structură simplificată pe trei niveluri care are ca scop să acopere majoritatea implementărilor și să ofere o mai bună vizibilitate a metodelor de integrare utilizate - Figura 16. Taxonomia pe trei straturi - Complet gestionat de Kubernetes, Kubernetes către HPC și HPC către Kubernetes - care definește forma de comunicare între planurile de control ale celor două infrastructuri distribuite este dublată de o altă axă dimensională, care reprezintă volatilitatea infrastructurii de găzduire - fixă/predefinită și furnizată la cerere. Această perspectivă de observație nuanțează oportunitățile de optimizare a costurilor în ceea ce privește consumul de resurse.

Îmbunătățirile sugerate în timpul analizei soluțiilor propun schimbări mici, dar cu impact în arhitectura și designul implementărilor găsite, în special în ceea ce privește înlocuirea obiectelor Kubernetes specifice în anumite cazuri de utilizare pentru optimizarea fluxurilor de lucru propuse de diverșii autori.

4.5. Prezentare generala a perspectivelor

După cum s-a putut observa în studiul de caz prezentat în secțiunea 4.3, simplificarea taxonomiei clasice de izolare a containerelor poate facilita aplicarea politicilor globale ale utilizatorilor. Pe deoparte, prin intermediul motorului în sine, dacă luăm în considerare doar activitatea pe topologii cu un singur nod. Pe de altă parte, prin intermediul unor orchestratori inteligenți, care pot manipula mai multe motoare și pot aplica politici de programare și constrângeri complexe asupra containerelor lansate. Prin urmare, o astfel de abordare prezintă o listă de oportunități de stabilire a limitelor asupra datelor sau procesatorilor de date, în contextul sistemelor distribuite critice.

Studiul de caz prezentat pe scurt în secțiunea 4.4, propune o taxonomie de clasificare pentru soluțiile găsite în literatura științifică recentă, ca modalitate de a sublinia posibilitatea de a combina diverse modele de guvernanță a sistemelor distribuite în scopul abordării unor scenarii complexe. Prin urmare, astfel de fuziuni pot îmbunătăți mecanismele native de planificare pentru a aborda problemele de localitate a datelor sau de distribuție a datelor. În capitolele anterioare, s-a demonstrat că paradigma cloud este destul de versatilă în ceea ce privește capacitățile de planificare, în timp ce sistemele de tip batch sunt foarte constrânse. Amestecarea celor două poate îmbunătăți capacitățile de susținere a

politicilor moderne de confidențialitate în ceea ce privește programarea datelor și a procesatorilor de date.

5. Evaluarea riscurilor

Atacurile cibernetice s-au intensificat în ultima vreme, deoarece oamenii tind să efectueze un spectru mai larg de operațiuni pe internet. În mod similar, utilizatorii rău intenționați și-au diversificat metodele de atac, proporțional cu noul val de utilizare a internetului. Pentru a depăși noile provocări în materie de securitate, este posibil ca serviciile care funcționează pe internet să fie nevoite să aplice propriul mecanism de protecție, pe baza unei analize calitative de securitate a activelor expuse. Rularea de honeypots alături de producție poate deveni un standard, deoarece, prin concepție, o astfel de configurație este capabilă să capteze informații comportamentale privind atacurile în timpul execuției. În plus, pe măsură ce adoptarea paradigmei microserviciilor a crescut, astfel de soluții tind să extindă suprafața de atac, prin urmare, devine relevantă validarea rezilienței activelor expuse în cazul unui atac cibernetic și generarea unor rapoarte de evaluare a riscurilor coerente și organice. Personalizarea politicilor de protecție, pe baza atributelor relevate de evaluarea riscurilor, poate reduce riscul de expunere a datelor confidențiale.

5.1. Întrebări de cercetare

- Care sunt oportunitățile de a genera feedback/rapoarte organice de securitate și confidențialitate privind activele/serviciile și infrastructurile care expun date critice?

5.2. Contribuții

- Bontaș, Carol-Sebastian, Ioan-Mihail Stan and Răzvan Rughiniș. "Honeypot generator using software defined networks and recursively defined topologies." *2022 21st RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2022
- *Stan, Ioan-Mihail, Ştefan-Dan Ciocîrlan and Răzvan Rughiniș. "Deployment methodology and practises for running Hybrid HoneyPots together with the microservice- based production over Kubernetes"*[10]

5.3. Studiu de caz - Generator de Honeypot pentru randomizarea modelelor de implementare

Rularea rețelelor Honeypot nu mai este o tendință emergentă, ci o necesitate din cauza numărului tot mai mare de exploatări. Astfel de configurații au scopul de a oferi o viziune coerentă asupra tehnicilor pe care le folosesc atacatorii cibernetici pentru a obține acces neautorizat la sisteme private. Prin urmare, implementarea de infrastructuri honeypot poate oferi mijloacele de a înțelege și de a preveni vulnerabilitățile de tip 0-day, tehnicile complexe de exploatare, comportamentul rău intenționat între clienții unei platforme și așa mai departe. O problemă generală a infrastructurilor honeypot este că acestea pot încorpora în structura lor amprenta dezvoltatorilor, prin urmare, instalările succesive ale unei infrastructuri similare în incinta a două organizații distincte pot dezvălui

[10] nu a fost publicat, rezumat și abstract acceptate de comisia de evaluare de la DS Symposium (http://doctorat.acs.pub.ro/ds-symposium-ro/)

scopul infrastructurii. Un infractor cibernetic, care atacă ambele configurații, poate observa modele comparabile care pot sugera că a fost prins în capcană pentru a fi observat și analizat.

Rețetele pentru desfășurarea unor astfel de infrastructuri complexe, alături de aplicațiile critice care trebuie protejate, sunt de obicei surse închise. Instituțiile guvernamentale sau întreprinderile tind să își mențină activele închise față de public, angajând specialiști pentru a furniza tehnici de apărare și configurații bazate pe standardele industriei. Cu toate acestea, în contextul actual, atacatorii găsesc metode rafinate de a ocoli standardele, în fiecare zi. Prin urmare, industria se află într-o cursă nedreaptă cu infractorii cibernetici pentru a înțelege provocările și a îmbunătăți convențiile de securitate și confidențialitate pentru a fi oferite publicului larg.

Având în vedere ambele provocări, propunerea mea este de a oferi o soluție open source pentru generarea de honeypots cu interacțiune ridicată, care poate servi atât ca un cadru de dezvoltare, cât și ca bază pentru construirea unor infrastructuri complexe de cercetare, cu un grad sporit de randomizare pentru a reduce problema similitudinii în cazul implementărilor succesive. Abordarea mea tinde spre generarea unor rapoarte coerente de securitate și confidențialitate a serviciilor care urmează să fie expuse publicului și spre crearea unor politici dedicate pentru a reduce riscurile asociate unui atac de succes. Combinată cu standardele sugerate de industrie, aceasta ar putea crește în mod semnificativ rezistența activelor la acțiuni rău intenționate. Soluția pentru generarea de honeypots, care poate încorpora un activ de producție înainte de lansare, este rentabilă, putând fi rulată ca un singur sistem și simulând, cu ajutorul containerelor, o întreagă infrastructură fizică. Caracterul open source al generatorului de honeypot-uri cu interacțiune ridicată oferă expunere către consultanți, specialiști sau dezvoltatorilor care pot contribui masiv la îmbunătățirea ulterioară a soluției. Sistemul dezvoltat încorporează concepte tehnologice precum Software Defined Networks (Rețele definite via Software), Recursively Defined Topologies (Topologii Definite Recursiv) și orchestrarea containerelor. Soluția propusă, care poate constitui un cadru pentru dezvoltarea de honeypots cu o suprafață de atac mare, introduce un formalism matematic pentru generarea de topologii recursive cu containere, un limbaj formal pentru configurarea parametrilor de scalabilitate asupra variabilelor formalismului matematic, un algoritm pentru construirea topologiilor și o metodologie proprie de orchestrare a containerelor. În același timp, studiul de caz propune o arhitectură extinsă, modernă, care combină soluțiile de rețele definite prin software (Software Defined Networks) cu o orchestrație proprietară asupra motoarelor de containerizare. Ambele facilitează lansarea și decomisionarea serviciilor vulnerabile, în mod dinamic, pe baza comportamentului atacatorului. Platforma dezvoltată poate fi utilizată în scopuri de cercetare, ca parte a efortului de contracarare a metodelor noi de atac, sau în incinta organizațiilor IT pentru a expune servicii și a simula producția. Înainte de a merge mai departe și de a prezenta arhitectura conceptului, este obligatoriu să se stabilească premisele și cerințele [193]:

În primul rând, am pus accentul pe honeypots cu interacțiune ridicată pentru complexitatea și relevanța lor în sectoarele critice și în cercetare. Prin urmare, acestea pot ajuta la stabilirea noilor tendințe în atacurile cibernetice și, de asemenea, pot dezvălui instrumentele utilizate [169]. În ceea ce privește honeypots cu interacțiune redusă, honeypots cu interacțiune ridicată necesită o infrastructură fizică sau virtuală, deoarece acționează ca rețele de producție legitime.

În al doilea rând, am luat în considerare problema dinamicii și ideea de a reduce amprenta de dezvoltare pentru a oferi legitimitate în contextul unui atac. Prin urmare, soluția trebuie să răspundă la situații de scalare imprevizibile și trebuie să fie capabilă să reacționeze fără asistență pe parcursul întregului ciclu de viață al unui atac sau al rulării unui honeypot. Soluția implementează și adaptează euristici topologice definite recursiv, cum ar fi FiConn[104] sau DCell[170], utilizate de obicei în construcția centrelor de date pentru ușurința lor în ceea ce privește scalarea verticală.

În al treilea rând, am învățat din tehnicile de construcție ale orchestratorilor de cloud și am oferit utilizatorilor capacitatea de a efectua modificări dinamice sau de a stabili cote de scalare încă de

la început. Prin urmare, serviciile containerizate vulnerabile și obiectele de rețea pot fi create sau șterse pe baza observațiilor. Heuristica inteligentă care implementează planul de control este reunită pentru a construi un orchestrator de containere proprietar și un cadru pentru dezvoltarea ulterioară.

Nu în ultimul rând, soluția trebuie să fie rezilientă la situații imprevizibile. Prin urmare, este important și obligatoriu să se pună la dispoziția inginerilor mecanisme de reacție rapidă și să poată modifica configurația rețelei sau să dezafecteze serviciile vulnerabile pentru a opri un atac complex. Cadrul propus se bazează pe rețelele definite prin software [171] și pe containerele Docker, care pot fi manipulate rapid și ușor prin intermediul unor servicii de gestionare proprietare.

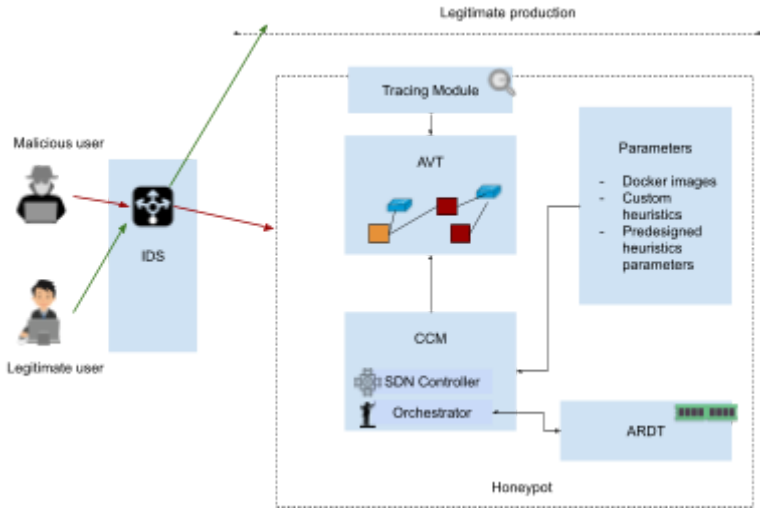

Figura 17 - Arhitectura concept generator honeypot

Arhitectura conceptului (Figura 17) cuprinde șase componente esențiale [193]:

- Sistemul de detectare a intruziunilor (IDS)
- Modulul de topologie abstractă definită în mod recursiv (Abstract Recursively Defined Topology - ARDT)
- Modul de urmărire (MD)
- Modulul de comandă centrală (CCM)
- Modulul de topologie virtuală activă (AVT)
- Generator de parametri

Sistemul de detectare a intruziunilor este interpus în infrastructura frontală și are sarcina de a efectua detectarea timpurie a intențiilor utilizatorului[172][173][174][175] în ceea ce privește serviciile expuse. Prin urmare, în cazul în care utilizatorul este rău intenționat, infrastructura dinamică a rețelei, furnizată prin intermediul controlorilor SDN și OpenVSwitch [176], va direcționa potențialul atacator către honeypot. Utilizatorii legitimi sunt certificați de același sistem și redirecționați către infrastructura de producție[193].

Modulul de urmărire are sarcina de a analiza comportamentul curent al utilizatorului rău intenționat în timp ce acesta este prins în interiorul honeypot-ului. Pentru simplificare, în versiunea inițială a sistemului propus, modulul de urmărire este capabil doar să detecteze și să analizeze atacurile de escaladare a privilegiilor, în care atacatorul reușește să exploateze vulnerabilități bine puse la punct pentru a obține o sesiune shell către containerele din apropiere[193].

Generatorul de honeypot cu interacțiune ridicată implementează strategia de evaluare leneșă[193]. Prin urmare, în loc să implementeze întreaga infrastructură a containerelor de la început, acesta lansează noi containere vulnerabile în timpul ciclului de viață al unui atac. În acest sens,

sistemul trebuie să aibă întotdeauna cunoștințe despre locația atacatorului și să genereze containere succesive în același timp în care vecinii sunt compromiși. O metodă simplificată de detectare a unor astfel de evenimente constă în capturarea acțiunilor de modificare a istoricul shell-ului unui container. Prin urmare, atunci când un container neatins este detectat ca fiind manipulat din interior, înseamnă că acesta a fost exploatat și că atacatorul se află în prezent în el. În același moment de timp, alte containere sunt lansate și o parte din containerele pornite anterior sunt scoase din uz, pe baza unei euristici simple, asemănătoare algoritmului minimax. Topologia activă curentă este reprezentată în arhitectură ca AVT. Structura logică (incluzând containerele active, dezafectate sau nelansate) este menținută în memorie în modulul Abstract Recursive Defined Topology. Astfel, planul de control are o reprezentare fizică și una logică și cunoaște în permanență ce container trebuie creat, reinstanțiat sau decomisionat în urma vizualizării imaginii virtuale a topologiei expuse și detectării comportamentului atacatorului[193].

Modulul central de comandă implementează funcția de orchestrare și adună informații sau interacționează cu alte module pentru a menține AVT într-o stare coerentă. Prin urmare, acesta interpretează parametrii de lansare de containere și euristica selectată, încapsulează controlerul SDN și funcția de orchestrare a containerelor și colectează informații relevante din ARDT. Generatorul de parametri funcționează atât ca un sistem de gestionare a configurației, cât și ca furnizor de euristici personalizate[193].

În plus, unul dintre obiectivele soluției propuse este acela de a putea implementa algoritmi de tip RDT, cum ar fi DCell[170] și FiConn[104], pentru a asigura scalabilitatea infrastructurii honeypot și, în același timp, pentru predictibilitate în gestionarea resurselor. Prin urmare, o parte din efortul de cercetare depus s-a concentrat pe găsirea corelațiilor dintre diferitele euristici. Prin intermediul sistemului de gestionare a configurației, se pot defini topologii de containere DCell sau FiConn sau alte euristici personalizate de tip RDT (Topologii Definite Recursiv) [193]. Efortul depus pentru a înțelege similitudinile dintre DCell și FiConn a condus la definirea unui formalism matematic simplificat pentru structurile RDT. Prin urmare, se poate lua în considerare următorul tuplu (N, K, c, fc) [193] unde

N -> numărul de sisteme de grad -1 (sisteme care fac parte din structura de bază, interconectate prin comutatoare)

K -> un grad maxim care trebuie atins în topologia RDT, prestabilit ca o măsură de predictibilitate și de gestionare a resurselor

c -> un grad minim de la care structura RDT impune existența unui număr fix de legături dinspre structurile cu grad c spre alte structuri similare sau de grad superior

fc -> funcția care furnizează numărul de legături pe care o structură de grad c trebuie să le aibă cu fiecare structură de grad superior

5.4. Studiu de caz - Construcția arhitecturilor Honeypots hibride pe Kubernetes

În ultimii ani, atacurile cibernetice au înregistrat o creștere semnificativă în ceea ce privește numărul și rata de succes. Un studiu condus de Accenture a arătat că, în 2021, numărul de atacuri per companie pe parcursul anului a crescut cu 31% față de 2020. Din aproximativ 270 de încercări de atacuri cibernetice, 29 dintre ele au reușit. Având în vedere varietatea metodelor de atac și creșterea numărului de vulnerabilități exploatabile, este mai greu să se scaleze șabloanele de securitate generice sau modelele de detectare a tiparelor de atac pentru a acoperi varietatea de servicii expuse în internet. Personalizarea unui astfel de proces necesită o înțelegere amplă a aspectelor de securitate ale activelor expuse. O metodă de obținere a informațiilor, cât mai rapid posibil, constă în analiza

comportamentului atacatorilor în premisele serviciului expus spre a fi consumat [181][182]. Rularea unui honeypot care plasează activele de producție într-un mediu controlat este o modalitate bună de a genera rapoarte de securitate organică per element. Apare o provocare considerabilă, atunci când atât producția, cât și rețeaua honeypot trebuie să fie afișate în mod transparent atât pentru utilizatorii legitimi, cât și pentru cei rău intenționați. Astfel, punctul de intrare spre infrastructură trebuie să fie împărțit între ambele ramuri ale infrastructurii: producție și honeypot. Eu numesc honeypot hibrid, infrastructura critică care găzduiește atât mediul de producție, cât și un honeypot cu interacțiune ridicată, în incinta căruia atacatorii trebuie să fie prinși în capcană, în timp ce ansamblul asigură în continuare funcția de producție.

Multe dintre aplicațiile de internet din zilele noastre se îndreaptă constant către paradigma microserviciilor. Conceptul prezentat de Martin Fowler și James Luis propune o modalitate de a sparge monoliții în servicii mici, independente, consumabile, axate pe furnizarea unei singure funcții minimale. Rulând împreună, acestea pot servi unor scopuri mai largi. Acei facilitatori care oferă mijloacele de a construi aplicații bazate pe arhitecturi de microservicii sunt motoarele de containerizare (de exemplu, Docker) și orchestratorii de containere (de exemplu, Kubernetes). Studiul actual se concentrează pe înțelegerea diverselor metodologii de lansare și a modelelor de proiectare, utilizate în astfel de tehnologii, cu scopul de a oferi un ghid pentru generarea de honeypot-uri hibride pentru soluții distribuite peste containere.

Acest studiu de caz propune o metodologie de proiectare a honeypot-urilor hibride bazată pe arhitectura microserviciilor, cu scopul de a proteja și izola activele legitime. Deciziile arhitecturale au o nuanță pragmatică și urmează ideea de a rula aplicații distribuite în containere pe un ecosistem Docker-Kubernetes. Studiul de caz prezintă atât arhitectura, cât și designul și introduce colecții de obiecte specifice Kubernetes, care pot ajuta la construcția unei taxonomii de izolare coerente. Evaluarea metodologiei observă mecanismele disponibile în ecosistemul Kubernetes care asigură izolarea multidimensională a activelor de producție față de cele găzduite în honeypots. Accentul se pune pe segregarea topologiei de comunicare și pe reziliența în contextul atacurilor de tip container-escape.

Într-o privire de ansamblu, metodologia propune 4 etape majore:

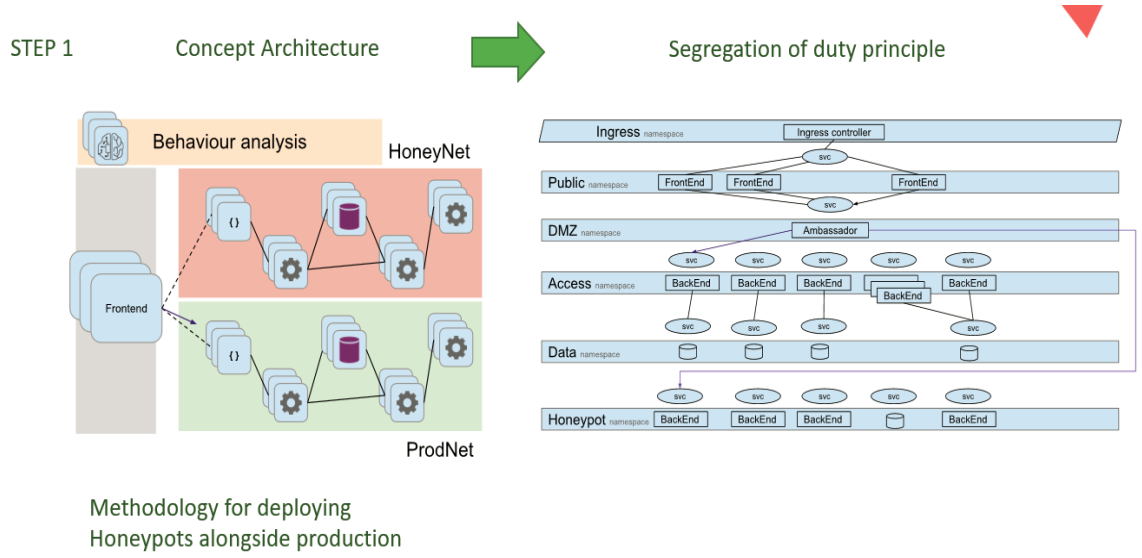

Figura 18 - Segregarea rolurilor serviciilor peste Kubernetes

Prima etapă propune o împărțire a microserviciilor sau a serviciilor containerizate în spații de nume diferite (namespace/workspace) cu scopul de a aplica politici de protecție distincte (Figura 18)

asupra activelor găzduite. Prin urmare, se pot separa serviciile publice/serviciile frontale de cele de backend şi de date, în timp ce o zonă demilitarizată este necesară pentru a îndeplini funcția de redirecționare către producția legitimă sau honeypot.

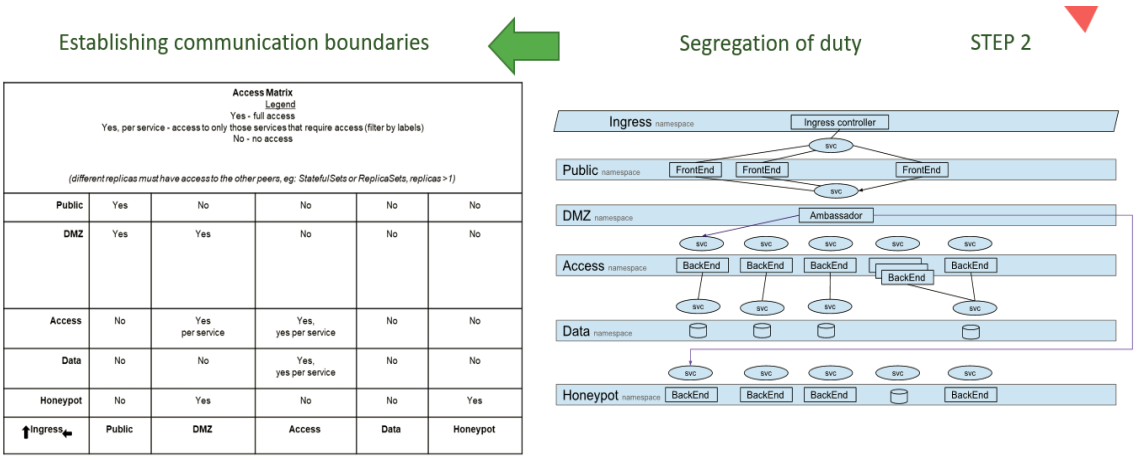

Figura 19 - Matricea de comunicare

A doua etapă (Figura 19) propune o matrice de comunicare şi acces pentru a defini limitele de comunicare pentru aplicația containerizată, cu respectarea principiilor arhitecturale din metodologia microserviciilor. Prin urmare, din punct de vedere tehnic, funcția de firewall statornic (stateful) livrată de diverşi facilitatori de rețea din Kubernetes, este consumată prin intermediul obiectelor native Kubernetes Network Policy.

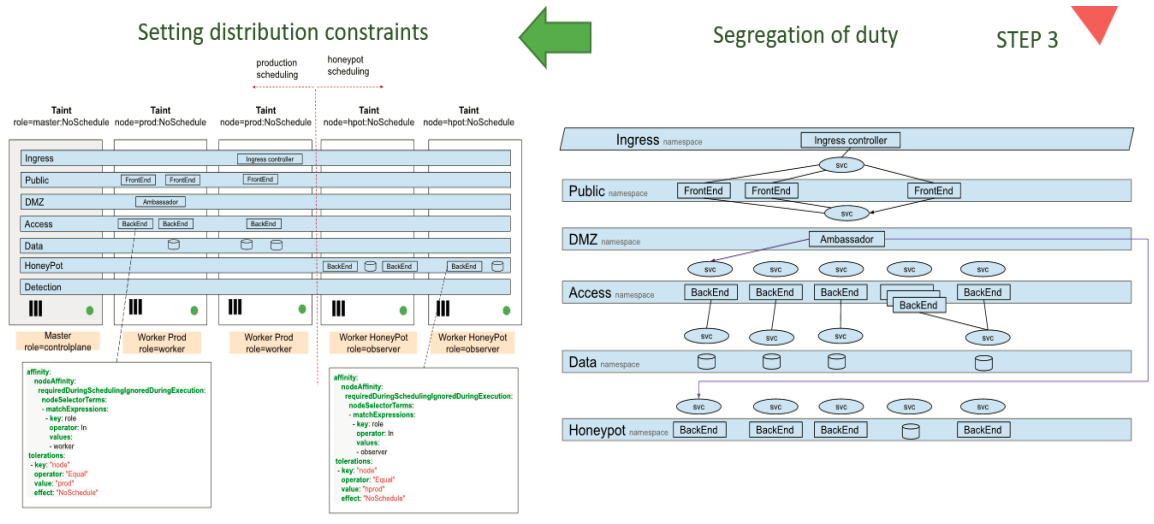

Figure 20 - Distributia serviciilor peste Kuberentes

A treia etapă (Figura 20) propune modul în care taxonomia de segregare poate fi distribuită pe clusterele Kubernetes, pentru a face posibilă aplicarea unor politici de protecție distincte pe infrastructura de bază. Prin urmare, funcția de afinitate este utilizată pentru a modifica

comportamentul implicit al planificatorului Kubernetes în scopul de a porni containere de aplicații specifice pe noduri dedicate.

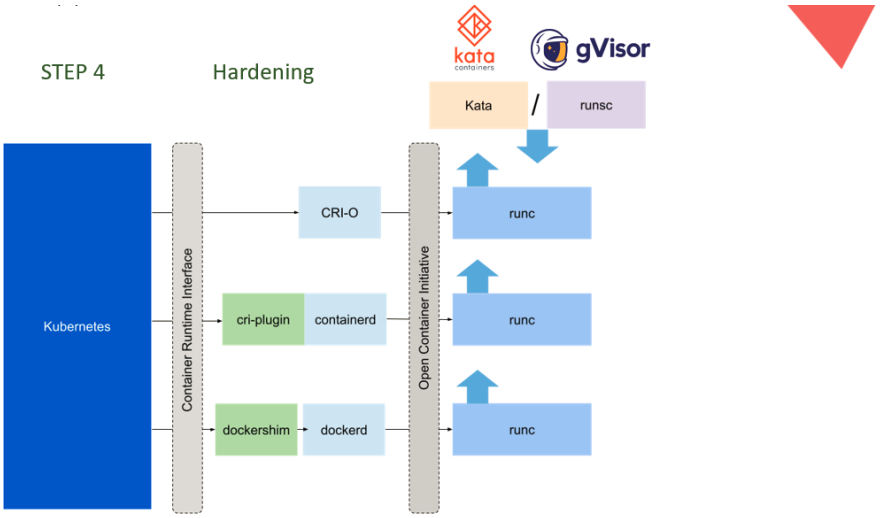

Figura 21 - Întărirea protecției sistemelor gazdă via motorul de containerizare

Cea de-a patra etapă (Figura 21) ia în considerare posibilitatea de a detașa modulul de execuție nativ al diferitelor motoare de containerizare și de a-l înlocui cu module de execuție mai sigure. Tehnologia Kata-containers înglobează fiecare container în mașini virtuale mici și ușoare, pentru a extinde stiva de sistem și pentru a externaliza API-ul pentru apelurile de sistem. gVisor cu runsc, oferă un controlor de admitere peste spațiul kernel existent, pentru a filtra orice posibil apel de sistem malițios, provenit din interiorul unui container.

5.5. Prezentare generala a perspectivelor

Un aspect relevant atunci când se implementează active critice este înțelegerea riscurilor de securitate care pot duce la expunerea datelor și, prin urmare, care ar putea să corupă granițele stabilite de obiectivele de confidențialitate a datelor. Prin urmare, construirea de "honeypots" în jurul activelor pentru detectarea timpurie a punctelor majore de intrare prin efracție poate constitui o necesitate reală, în special în cazul infrastructurilor critice. Prin urmare, se pot expune versiuni timpurii ale serviciilor în honeypots, înainte de a le muta în producție, în timp ce, în cazuri particulare, ambele ramuri ale infrastructurii pot coexista.

În studiul de caz prezentat în secțiunea 5.3, soluția propusă a reușit să stabilească un mix între două concepte-cheie moderne în furnizarea dinamică de servicii și topologiile scalabile: Software Defined Network (Rețeaua definită prin software) și Recursively Defined Topologies (Topologii definite în mod recursiv). Prin intermediul platformei furnizate, concepută ca un șablon, cercetătorii și dezvoltatorii pot genera honeypots cu interacțiune ridicată, cu scopul de a contracara exploatările de tip 0-day și de a înțelege noile tehnici de atac cibernetic. Prin versatilitatea sa și prin faptul că este concepută ca o infrastructură containerizată peste Docker, devine mai ușoară implementarea de scenarii personalizate/variabile și expunerea rapidă a acestora către exterior în infrastructuri cu un singur nod.

În studiul de caz prezentat în secțiunea 5.4, lucrarea oferă o metodologie care constituie piatra de temelie pentru construirea de arhitecturi honeypot hibride peste aplicații bazate pe microservicii. Abordarea de jos în sus absoarbe modele și metode de proiectare din tehnologii precum Kubernetes și Docker, în timp ce acestea furnizează, de asemenea, infrastructura de bază pentru studiul de caz

realizat. Metodologia prezintă două viziuni arhitecturale principale: topologia de comunicare și stiva de securitate implementată peste nodurile gazdă. De asemenea, arată cum pot fi integrate rețele honeypot în ecosistemul de producție, într-o configurație hibridă.

6. Concluzii

Lucrările descrise în teza de față oferă perspective relevante în ceea ce privește problema confidențialității în infrastructuri complexe și distribuite, cu accent pe implementările critice și costisitoare. Teza aduce în prim-plan mai multe modele de guvernanță asupra sistemelor distribuite, analizând capacitatea acestora de a răspunde la problemele actuale de securitate și confidențialitate. De asemenea, se discută modul în care tehnologiile emergente identificate pe piață, utilizate în forma lor originală sau adaptate, pot avea un impact major asupra menținerii confidențialității datelor. Astfel, am prezentat modul în care blockchain poate completa funcția de comunicare, asigurând un mediu coerent pentru cooperarea anonimă. În același timp, a fost luată în considerare necesitatea construirii de infrastructuri honeypot în jurul activelor de producție, pentru o mai bună înțelegere a punctelor vulnerabile în cazul unei expuneri publice.

Teza abordează cinci puncte focale majore în ceea ce privește analiza aspectelor legate de confidențialitatea datelor: accesibilitatea datelor, anonimitatea/anonimizarea datelor, transportul datelor, distribuția și orchestrarea datelor și evaluarea riscurilor asupra datelor expuse. De asemenea, prezint diverse studii de caz, dezvoltate în jurul unor infrastructuri critice, vizând sectoare precum cel guvernamental, financiar sau cel al tehnologiei informației.

Bibliografie

[1] Feldmann, Anja, et al. "The lockdown effect: Implications of the COVID-19 pandemic on internet traffic." *Proceedings of the ACM internet measurement conference*. 2020.

[2] Hijji, Mohammad, and Gulzar Alam. "A multivocal literature review on growing social engineering based cyber-attacks/threats during the COVID-19 pandemic: challenges and prospective solutions." *Ieee Access* 9 (2021): 7152-7169.

[3] Rinaldi, Steven M., James P. Peerenboom, and Terrence K. Kelly. "Identifying, understanding, and analyzing critical infrastructure interdependencies." *IEEE control systems magazine* 21.6 (2001): 11-25.

[4] Moteff, John, and Paul Parfomak. "Critical infrastructure and key assets: definition and identification." Library of Congress Washington DC Congressional Research Service, 2004.

[5] Forti, Alessandra, et al. "The fight against COVID-19: Running Folding@ Home simulations on ATLAS resources." *EPJ Web of Conferences*. Vol. 251. EDP Sciences, 2021.

[6] Foster, Ian, et al. "The grid2003 production grid: Principles and practice." *Proceedings. 13th IEEE International Symposium on High performance Distributed Computing, 2004.*. IEEE, 2004.

[7] Cesini, Daniele, et al. "The eXtreme-DataCloud project: data management services for the next generation distributed e-infrastructures." *2018 Conference Grid, Cloud & High Performance Computing in Science (ROLCG)*. IEEE, 2018.

[8] Koops, Bert-Jaap. "The trouble with European data protection law." *International data privacy law* 4.4 (2014): 250-261.

[9] Houser, Kimberly A., and W. Gregory Voss. "GDPR: The end of Google and Facebook or a new paradigm in data privacy." *Rich. JL & Tech.* 25 (2018): 1.

[10] Isaak, Jim, and Mina J. Hanna. "User data privacy: Facebook, Cambridge Analytica, and privacy protection." *Computer* 51.8 (2018): 56-59.

[11] Nam, Yeonghun, et al. "Global-scale GDPR Compliant Data Sharing System." *2020 International Conference on Electronics, Information, and Communication (ICEIC)*. IEEE, 2020.

[12] Politou, Eugenia, et al. "Backups and the right to be forgotten in the GDPR: An uneasy relationship." *Computer Law & Security Review* 34.6 (2018): 1247-1257.

[13] Xiao, Zhifeng, and Yang Xiao. "Security and privacy in cloud computing." IEEE communications surveys & tutorials 15.2 (2012): 843-859.

[14] Duan, Yucong, et al. "Everything as a service (XaaS) on the cloud: origins, current and future trends." *2015 IEEE 8th International Conference on Cloud Computing*. IEEE, 2015

[15] Höfer, C. N., and Georgios Karagiannis. "Cloud computing services: taxonomy and comparison." *Journal of Internet Services and Applications* 2.2 (2011): 81-94.
[16] Lu, Qinghua, et al. "uBaaS: A unified blockchain as a service platform." *Future Generation Computer Systems* 101 (2019): 564-575.
[17] Andersen, Michael P., Gabe Fierro, and David E. Culler. "Enabling synergy in iot: Platform to service and beyond." *Journal of Network and Computer Applications* 81 (2017): 96-110.
[18] Zhao, Feng, Chao Li, and Chun Feng Liu. "A cloud computing security solution based on fully homomorphic encryption." *16th international conference on advanced communication technology*. IEEE, 2014.
[19] de Castro, Leo, et al. "Does Fully Homomorphic Encryption Need Compute Acceleration?." *arXiv preprint arXiv:2112.06396* (2021).
[20] Merkel, Dirk. "Docker: lightweight linux containers for consistent development and deployment." *Linux j* 239.2 (2014): 2.
[21] De Lauretis, Lorenzo. "From monolithic architecture to microservices architecture." *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. IEEE, 2019.
[22] Jacob, Bart, et al. "Introduction to grid computing." *IBM redbooks* (2005): 3-6.
[23] Habib, Muhammad Asif, and Michael Thomas Krieger. "Security in Grid Computing." *Seminar aus Netzwerke und Sicherheit: Communication Infrastructure*. 2008.
[24] Foster, Ian, et al. "Cloud computing and grid computing 360-degree compared." *2008 grid computing environments workshop*. Ieee, 2008.
[25] Dowd, Kevin, and Charles Severance. "High performance computing." (2010).
[26] Zhang, Di, et al. "RLScheduler: an automated HPC batch job scheduler using reinforcement learning." *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2020.
[27] Mu'alem, Ahuva W., and Dror G. Feitelson. "Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling." *IEEE transactions on parallel and distributed systems* 12.6 (2001): 529-543.
[28] Courtès, Ludovic, and Ricardo Wurmus. "Reproducible and user-controlled software environments in HPC with Guix." *European Conference on Parallel Processing*. Springer, Cham, 2015.
[29] Sotiriadis, Stelios, et al. "From meta-computing to interoperable infrastructures: A review of meta-schedulers for HPC, grid and cloud." *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*. IEEE, 2012.
[30] Di Pierro, Massimo. "What is the blockchain?." *Computing in Science & Engineering* 19.5 (2017): 92-95.
[31] Sankar, Lakshmi Siva, M. Sindhu, and M. Sethumadhavan. "Survey of consensus protocols on blockchain applications." *2017 4th international conference on advanced computing and communication systems (ICACCS)*. IEEE, 2017.
[32] Rathod, Nidhee, and Dilip Motwani. "Security threats on blockchain and its countermeasures." *Int. Res. J. Eng. Technol* 5.11 (2018): 1636-1642.
[33] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." *Decentralized Business Review* (2008): 21260.
[34] Gervais, Arthur, et al. "On the security and performance of proof of work blockchains." *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016.
[35] Tanwar, Sudeep, Karan Parekh, and Richard Evans. "Blockchain-based electronic healthcare record system for healthcare 4.0 applications." *Journal of Information Security and Applications* 50 (2020): 102407.
[36] Androulaki, Elli, et al. "Hyperledger fabric: a distributed operating system for permissioned blockchains." *Proceedings of the thirteenth EuroSys conference*. 2018.
[37] Zhang, Jian, et al. "Deploying blockchain technology in the supply chain." *Computer security threats* (2019): 57.
[38] Ferdousi, Tanvir, Don Gruenbacher, and Caterina M. Scoglio. "A permissioned distributed ledger for the US beef cattle supply chain." *IEEE Access* 8 (2020): 154833-154847.
[39] Metcalfe, William. "Ethereum, smart contracts, DApps." *Blockchain and Crypt Currency* (2020): 77.
[40] Ayeni, O. A., B. K. Alese, and L. O. Omotosho. "Design and implementation of a medium interaction honeypot." *International Journal of Computer Applications* 975 (2013): 8887.
[41] Saputro, Elang Dwi, Yudha Purwanto, and Muhammad Faris Ruriawan. "Medium interaction honeypot infrastructure on the internet of things." *2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*. IEEE, 2021.
[42] Deogirikar, Jyoti, and Amarsinh Vidhate. "Security attacks in IoT: A survey." *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 2017.
[43] Elmsheuser, Johannes, and Alessandro Di Girolamo. "Overview of the ATLAS distributed computing system." *EPJ Web of Conferences*. Vol. 214. EDP Sciences, 2019.
[44] Benjamin, D., A. Filipcic, and A. Klimentov. "ATLAS HPC Data Processing and Simulation."
[45] Fowler, Martin, and Matthew Foemmel. "Continuous integration." (2006).
[46] Ferrara, Pietro, and Fausto Spoto. "Static Analysis for GDPR Compliance." *ITASEC*. 2018.

[47] Lossent, Alexandre, A. Rodriguez Peon, and A. Wagner. "PaaS for web applications with OpenShift Origin." *Journal of Physics: Conference Series*. Vol. 898. No. 8. IOP Publishing, 2017.
[48] Ratis, Pavlos, Senior Site Reliability Engineer, and Red Hat. "Lessons Learned Using the Operator Pattern to Build a Kubernetes Platform." (2021).
[49] Schmeling, Benjamin, and Maximilian Dargatz. "Operations As Code with Kubernetes Operators and GitOps." *Kubernetes Native Development*. Apress, Berkeley, CA, 2022. 303-385.
[50] Mahboob, Jamal, and Joel Coffman. "A Kubernetes CI/CD Pipeline with Asylo as a Trusted Execution Environment Abstraction Framework." *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2021.
[51] Caban, William. "Architecting OpenShift Jenkins Pipelines." Architecting and Operating OpenShift Clusters. Apress, Berkeley, CA, 2019. 195-220.
[52] Ebert, Christof, et al. "DevOps." *Ieee Software* 33.3 (2016): 94-100.
[53] Kumar, Rakesh, and Rinkaj Goyal. "When Security Meets Velocity: Modeling Continuous Security for Cloud Applications using DevSecOps." *Innovative Data Communication Technologies and Application*. Springer, Singapore, 2021. 415-432.
[54] Beetz, Florian, and Simon Harrer. "GitOps: The Evolution of DevOps?." *IEEE Software* (2021).
[55] Barreiro, Fernando, et al. "The Future of Distributed Computing Systems in ATLAS: Boldly Venturing Beyond Grids." *EPJ Web of Conferences*. Vol. 214. EDP Sciences, 2019.
[56] Lukas, Wolfgang. "Fast simulation for ATLAS: Atlfast-II and ISF." *Journal of Physics: Conference Series*. Vol. 396. No. 2. IOP Publishing, 2012.
[57] Klimentov, A. A. "Exascale Data Processing in Heterogeneous Distributed Computing Infrastructure for Applications in High Energy Physics." *Physics of Particles and Nuclei* 51.6 (2020): 995-1068.
[58] Svirin, Pavlo, et al. "BigPanDA: PanDA Workload Management System and its Applications beyond ATLAS." EPJ Web of Conferences. Vol. 214. EDP Sciences, 2019.
[59] Korchuganova, Tatiana, et al. *The ATLAS BigPanDA Monitoring System Architecture*. No. ATL-SOFT-PROC-2018-019. ATL-COM-SOFT-2018-167, 2018.
[60] Stan, Ioan-Mihail, Siarhei Padolski, and Christopher Jon Lee. "Exploring the self-service model to visualize the results of the ATLAS Machine Learning analysis jobs in BigPanDA with OpenShift OKD3." *EPJ Web of Conferences*. Vol. 251. EDP Sciences, 2021.
[61] Beltre, Angel M., et al. "Enabling HPC workloads on cloud infrastructure using Kubernetes container orchestration mechanisms." *2019 IEEE/ACM International Workshop on Containers and New Orchestration Paradigms for Isolated Environments in HPC (CANOPIE-HPC)*. IEEE, 2019.
[62] Orzechowski, Michal, et al. "Transparent deployment of scientific workflows across clouds-kubernetes approach." *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*. IEEE, 2018.
[63] Bahadori, Kiyana, and Tullio Vardanega. "DevOps meets dynamic orchestration." *International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment*. Springer, Cham, 2018.
[64] ATLAS Distributed Computing, https://twiki.cern.ch/twiki/bin/viewauth/ AtlasComputing/AtlasDistributedComputing (2021), accessed: 2021-01-28
[65] OKD - The Community Distribution of Kubernetes that powers Red Hat OpenShift, https://www.okd.io/ (2021), accessed: 2021-06-17
[66] Alekseev, A., et al. "ATLAS BigPanDA monitoring." *Journal of Physics: Conference Series*. Vol. 1085. No. 3. IOP Publishing, 2018.
[67] Padolski, S., et al. "Data visualization and representation in ATLAS BigPanDA monitoring." *Научная визуализация* 10.1 (2018): 69-76.
[68] Vukotic, Ilija, et al. *Atlas analytics and machine learning platforms*. No. ATL-SOFT-SLIDE-2018-417. ATL-COM-SOFT-2018-084, 2018.
[69] Radovic, Alexander, et al. "Machine learning at the energy and intensity frontiers of particle physics." *Nature* 560.7716 (2018): 41-48.
[70] Campana, Simone, and ATLAS collaboration. "ATLAS Distributed Computing in LHC Run2." *Journal of Physics: Conference Series*. Vol. 664. No. 3. IOP Publishing, 2015.
[71] Zaharia, Matei, et al. "Accelerating the machine learning lifecycle with MLflow." *IEEE Data Eng. Bull.* 41.4 (2018): 39-45.
[72] Serfon, Cedric, et al. "Rucio, the next-generation Data Management system in ATLAS." *Nuclear and particle physics proceedings* 273 (2016): 969-975.
[73] Burns, Brendan, and David Oppenheimer. "Design patterns for container-based distributed systems." *8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 16)*. 2016.

[74] Berghaus, Frank, et al. "Federating distributed storage for clouds in ATLAS." *Journal of Physics: Conference Series*. Vol. 1085. No. 3. IOP Publishing, 2018.
[75] Monreale, Anna, et al. "Movement data anonymity through generalization." *Trans. Data Priv.* 3.2 (2010): 91-121.
[76] Samarati, Pierangela, and Latanya Sweeney. "Generalizing data to provide anonymity when disclosing information." *PODS*. Vol. 98. No. 188. 1998.
[77] Zhang, Xuyun, et al. "An efficient quasi-identifier index based approach for privacy preservation over incremental data sets on cloud." *Journal of Computer and System Sciences* 79.5 (2013): 542-555
[78] Motwani, Rajeev, and Ying Xu. "Efficient algorithms for masking and finding quasi-identifiers." *Proceedings of the Conference on Very Large Data Bases (VLDB)*. 2007.
[79] Mansour, Huda O., et al. "Quasi-Identifier recognition algorithm for privacy preservation of cloud data based on risk reidentification." *Wireless Communications and Mobile Computing* 2021 (2021).
[80] Feng, Qi, et al. "A survey on privacy protection in blockchain system." *Journal of Network and Computer Applications* 126 (2019): 45-58.
[81] Gupta, Suyash, and Mohammad Sadoghi. "Blockchain transaction processing." *arXiv preprint arXiv:2107.11592* (2021).
[82] Al-Breiki, Hamda, et al. "Trustworthy blockchain oracles: review, comparison, and open research challenges." *IEEE Access* 8 (2020): 85675-85685.
[83] Zheng, Zibin, et al. "An overview on smart contracts: Challenges, advances and platforms." *Future Generation Computer Systems* 105 (2020): 475-491.
[84] Mense, Alexander, and Markus Flatscher. "Security vulnerabilities in ethereum smart contracts." *Proceedings of the 20th international conference on information integration and web-based applications & services*. 2018.
[85] Oliveira, Luis, et al. "To token or not to token: Tools for understanding blockchain tokens." (2018).
[86] Shirole, Mahesh, Maneesh Darisi, and Sunil Bhirud. "Cryptocurrency token: An overview." *IC-BCT 2019* (2020): 133-140.
[87] Davydov, Vyacheslav, et al. "Token standard for heterogeneous assets digitization into commodity." *Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications*. 2019.
[88] Victor, Friedhelm, and Bianca Katharina Lüders. "Measuring ethereum-based erc20 token networks." *International Conference on Financial Cryptography and Data Security*. Springer, Cham, 2019.
[89] Brünjes, Lars, and Murdoch J. Gabbay. "UTxO-vs account-based smart contract blockchain programming paradigms." *International Symposium on Leveraging Applications of Formal Methods*. Springer, Cham, 2020.
[90] Casale-Brunet, Simone, et al. "Networks of Ethereum non-fungible Tokens: a graph-based analysis of the ERC-721 ecosystem." *2021 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2021.
[91] Muthe, Koushik Bhargav, Khushboo Sharma, and Karthik Epperla Nagendra Sri. "A blockchain based decentralized computing and NFT infrastructure for game networks." *2020 Second International Conference on Blockchain Computing and Applications (BCCA)*. IEEE, 2020.
[92] Stan, Ioan-Mihail, Ilie-Constantin Barac, and Daniel Rosner. "Architecting a scalable e-election system using Blockchain technologies." *2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2021.
[93] Cortier, Véronique, and Cyrille Wiedling. "A formal analysis of the Norwegian E-voting protocol." *International Conference on Principles of Security and Trust*. Springer, Berlin, Heidelberg, 2012.
[94] Alvarez, R. Michael, Thad E. Hall, and Alexander H. Trechsel. "Internet voting in comparative perspective: the case of Estonia." *PS: Political Science & Politics* 42.3 (2009): 497-505.
[95] Curran, Kevin. "E-Voting on the Blockchain." *The Journal of the British Blockchain Association* 1.2 (2018): 4451.
[96] Sajana, P., M. Sindhu, and M. Sethumadhavan. "On blockchain applications: hyperledger fabric and ethereum." *International Journal of Pure and Applied Mathematics* 118.18 (2018): 2965-2970.
[97] Adiputra, Cosmas Krisna, Rikard Hjort, and Hiroyuki Sato. "A proposal of blockchain-based electronic voting system." *2018 second world conference on smart trends in systems, security and sustainability (WorldS4)*. IEEE, 2018.
[98] Shi, Yichun, and Anil K. Jain. "DocFace+: ID document to selfie matching." *IEEE Transactions on Biometrics, Behavior, and Identity Science* 1.1 (2019): 56-67.
[99] Imerman, Michael B., and Frank J. Fabozzi. "Cashing in on innovation: a taxonomy of FinTech." *Journal of Asset Management* 21.3 (2020): 167-177.
[100] Fatrah, Aicha, et al. "Proof of concept blockchain-based voting system." *Proceedings of the 4th International Conference on Big Data and Internet of Things*. 2019.
[101] Thakkar, Parth, Senthil Nathan, and Balaji Viswanathan. "Performance benchmarking and optimizing hyperledger fabric blockchain platform." *2018 IEEE 26th international symposium on modeling, analysis, and simulation of computer and telecommunication systems (MASCOTS)*. IEEE, 2018.
[102] Al-Fares, Mohammad, Alexander Loukissas, and Amin Vahdat. "A scalable, commodity data center network architecture." *ACM SIGCOMM computer communication review* 38.4 (2008): 63-74.

[103] Lebiednik, Brian, Aman Mangal, and Niharika Tiwari. "A survey and evaluation of data center network topologies." *arXiv preprint arXiv:1605.01701* (2016).
[104] Li, Dan, et al. "FiConn: Using backup port for server interconnection in data centers." *IEEE INFOCOM 2009*. IEEE, 2009.
[105] Guo, Chuanxiong, et al. "BCube: a high performance, server-centric network architecture for modular data centers." *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. 2009.
[106] Deshpande, Varun, Hakim Badis, and Laurent George. "Efficient topology control of blockchain peer to peer network based on SDN paradigm." *Peer-to-Peer Networking and Applications* 15.1 (2022): 267-289.
[107] Zhang, Jiboning. "Interaction design research based on large data rule mining and blockchain communication technology." *Soft Computing* 24.21 (2020): 16593-16604.
[108] Spasovski, Jason, and Peter Eklund. "Proof of stake blockchain: performance and scalability for groupware communications." *Proceedings of the 9th International Conference on Management of Digital EcoSystems*. 2017.
[109] Göbel, Johannes, and Anthony E. Krzesinski. "Increased block size and Bitcoin blockchain dynamics." *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, 2017.
[110] Vujičić, Dejan, Dijana Jagodić, and Siniša Ranđić. "Blockchain technology, bitcoin, and Ethereum: A brief overview." *2018 17th international symposium infoteh-jahorina (infoteh)*. IEEE, 2018.
[111] Reyna, Ana, et al. "On blockchain and its integration with IoT. Challenges and opportunities." *Future generation computer systems* 88 (2018): 173-190.
[112] Samaniego, Mayra, Uurtsaikh Jamsrandorj, and Ralph Deters. "Blockchain as a Service for IoT." *2016 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData)*. IEEE, 2016.
[113] Dannen, Chris. *Introducing Ethereum and solidity*. Vol. 1. Berkeley: Apress, 2017.
[114] Antonopoulos, Andreas M., and Gavin Wood. *Mastering ethereum: building smart contracts and dapps*. O'reilly Media, 2018.
[115] Moubarak, Joanna, Eric Filiol, and Maroun Chamoun. "On blockchain security and relevant attacks." *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*. IEEE, 2018.
[116] Siddiqui, Shams Tabrez, et al. "Blockchain security threats, attacks and countermeasures." *Ambient Communications and Computer Systems*. Springer, Singapore, 2020. 51-62.
[117] Aggarwal, Shubhani, and Neeraj Kumar. "Attacks on blockchain." Advances in Computers. Vol. 121. Elsevier, 2021. 399-410.
[118] Huang, Dongyan, Xiaoli Ma, and Shengli Zhang. "Performance analysis of the raft consensus algorithm for private blockchains." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50.1 (2019): 172-181.
[119] Popa, Alin Bogdan, Ioan Mihail Stan, and Răzvan Rughiniș. "Instant payment and latent transactions on the Ethereum Blockchain." *2018 17th RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2018.
[120] Wood, Ethereum. "A secure decentralised generalised transaction ledger, Ethereum Proj." *Yellow Pap* 151: 1.
[121] Vogelsteller, Fabian, and Vitalik Buterin. "ERC-20 token standard." *Ethereum Foundation (Stiftung Ethereum), Zug, Switzerland* (2015).
[122] Helliar, Christine V., et al. "Permissionless and permissioned blockchain diffusion." *International Journal of Information Management* 54 (2020): 102136.
[123] Liu, Manlu, Kean Wu, and Jennifer Jie Xu. "How will blockchain technology impact auditing and accounting: Permissionless versus permissioned blockchain." *Current Issues in auditing* 13.2 (2019): A19-A29.
[124] Cash, Michael, and Mostafa Bassiouni. "Two-tier permission-ed and permission-less blockchain for secure data sharing." *2018 IEEE International Conference on Smart Cloud (SmartCloud)*. IEEE, 2018.
[125] Jang, Hyeji, Sung H. Han, and Ju Hwan Kim. "User perspectives on blockchain technology: user-centered evaluation and design strategies for dapps." *IEEE Access* 8 (2020): 226213-226223.
[126] Truong, Nguyen Binh, et al. "Gdpr-compliant personal data management: A blockchain-based solution." *IEEE Transactions on Information Forensics and Security* 15 (2019): 1746-1761.
[127] Schoo, Peter, et al. "Challenges for cloud networking security." *International Conference on Mobile Networks and Management*. Springer, Berlin, Heidelberg, 2010.
[128] Zhao, Zhifeng, Feng Hong, and Rongpeng Li. "SDN based VxLAN optimization in cloud computing networks." *IEEE Access* 5 (2017): 23312-23319.
[129] Benisi, Nazanin Zahed, Mehdi Aminian, and Bahman Javadi. "Blockchain-based decentralized storage networks: A survey." *Journal of Network and Computer Applications* 162 (2020): 102656.
[130] Barenji, Ali Vatankhah, et al. "Blockchain-based cloud manufacturing: Decentralization." *arXiv preprint arXiv:1901.10403* (2019).
[131] Ruan, Bowen, et al. "A performance study of containers in cloud environment." *Asia-Pacific Services Computing Conference*. Springer, Cham, 2016.

[132] Casalicchio, Emiliano. "Container orchestration: a survey." *Systems Modeling: Methodologies and Tools* (2019): 221-235.
[133] De Lucia, Michael J. *A survey on security isolation of virtualization, containers, and unikernels*. US Army Research Laboratory Aberdeen Proving Ground United States, 2017.
[134] Rosen, Rami. "Resource management: Linux kernel namespaces and cgroups." *Haifux, May* 186 (2013): 70.
[135] Stan, Ioan-Mihail, Daniel Rosner, and Ştefan-Dan Ciocîrlan. "Enforce a global security policy for user access to clustered container systems via user namespace sharing." *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2020.
[136] Sun, Yuqiong, et al. "Security namespace: making linux security frameworks available to containers." *27th USENIX Security Symposium (USENIX Security 18)*. 2018.
[137] Dimou, Fani. "Automatic security hardening of Docker containers using Mandatory Access Control, specialized in defending isolation." (2019).
[138] Ghavamnia, Seyedhamed, et al. "Confine: Automated system call policy generation for container attack surface reduction." *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*. 2020.
[139] Biederman, Eric W., and Linux Networx. "Multiple instances of the global linux namespaces." *Proceedings of the Linux Symposium*. Vol. 1. No. 1. Citeseer, 2006.
[140] Jian, Zhiqiang, and Long Chen. "A defense method against docker escape attack." *Proceedings of the 2017 International Conference on Cryptography, Security and Privacy*. 2017.
[141] Lin, Xin, et al. "A measurement study on linux container security: Attacks and countermeasures." *Proceedings of the 34th Annual Computer Security Applications Conference*. 2018.
[142] Llopis, Pablo, et al. "Integrating HPC into an agile and cloud-focused environment at CERN." *EPJ Web of Conferences*. Vol. 214. EDP Sciences, 2019.
[143] Priedhorsky, Reid, and Tim Randles. "Charliecloud: Unprivileged containers for user-defined software stacks in hpc." *Proceedings of the international conference for high performance computing, networking, storage and analysis*. 2017.
[144] Bui, Thanh. "Analysis of docker security." *arXiv preprint arXiv:1501.02967* (2015).
[145] Kurtzer, Gregory M., Vanessa Sochat, and Michael W. Bauer. "Singularity: Scientific containers for mobility of compute." *PloS one* 12.5 (2017): e0177459.
[146] Le, Emily, and David Paz. "Performance analysis of applications using singularity container on sdsc comet." *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact*. 2017. 1-4.
[147] Culic, Ioana-Maria, and Alexandru Radovici. "Development platform for building advanced Internet of Things systems." *2017 16th RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2017.
[148] Florea, Iulia, Laura Cristina Ruse, and Razvan Rughinis. "Challenges in security in Internet of Things." *2017 16th RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2017.
[149] Rad, Babak Bashari, Harrison John Bhatti, and Mohammad Ahmadi. "An introduction to docker and analysis of its performance." *International Journal of Computer Science and Network Security (IJCSNS)* 17.3 (2017): 228
[150] Benedicic, Lucas, et al. "Sarus: Highly scalable Docker containers for HPC systems." *International Conference on High Performance Computing*. Springer, Cham, 2019.
[151]. Wang, Xingyu, Junzhao Du, and Hui Liu. "Performance and isolation analysis of RunC, gVisor and Kata Containers runtimes." *Cluster Computing* 25.2 (2022): 1497-1513.
[152]. Mavridis, Ilias, and Helen Karatza. "Orchestrated sandboxed containers, unikernels, and virtual machines for isolation‐enhanced multitenant workloads and serverless computing in cloud." *Concurrency and Computation: Practice and Experience* (2021): e6365.
[153]. Megino, Fernando Harald Barreiro, et al. "Using Kubernetes as an ATLAS computing site." *EPJ Web of Conferences*. Vol. 245. EDP Sciences, 2020.
[154]. Ferreira, Arnaldo Pereira, and Richard Sinnott. "A performance evaluation of containers running on managed kubernetes services." *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2019.
[155]. Wei-guo, Zhang, Ma Xi-lin, and Zhang Jin-zhong. "Research on kubernetes' resource scheduling scheme." *Proceedings of the 8th International Conference on Communication and Network Security*. 2018.
[156]. Cérin, Christophe, Nicolas Greneche, and Tarek Menouer. "Towards pervasive containerization of HPC job schedulers." *2020 IEEE 32nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. IEEE, 2020.
[157].Piras, Marco Enrico, et al. "Container orchestration on HPC clusters." *International Conference on High Performance Computing*. Springer, Cham, 2019.
[158]. Colonnelli, Iacopo, et al. "StreamFlow: cross-breeding cloud with HPC." *IEEE Transactions on Emerging Topics in Computing* 9.4 (2020): 1723-1737.

[159]. López-Huguet, Sergio, et al. "Seamlessly managing HPC workloads through Kubernetes." *International Conference on High Performance Computing*. Springer, Cham, 2020.
[160]. Stan, Ioan-Mihail et al. *Exploring the self-service model to visualize the results of the ATLAS Machine Learning analysis jobs in BigPanDA with OpenShift OKD3*. CERN, 2021
[161]. Zhou, Naweiluo, et al. "Container orchestration on HPC systems." *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*. IEEE, 2020.
[162]. Zhou, Naweiluo, et al. "Container orchestration on HPC systems through Kubernetes." *Journal of Cloud Computing* 10.1 (2021): 1-14.
[163]. Dakic, Vedran, Mario Kovac, and Jasmin Redzepagic. "OPTIMIZING KUBERNETES PERFORMANCE, EFFICIENCY AND ENERGY FOOTPRINT IN HETEROGENOUS HPC ENVIRONMENTS." *Annals of DAAAM & Proceedings* 10.2 (2021).
[164] Dragoni, Nicola, et al. "Microservices: yesterday, today, and tomorrow." *Present and ulterior software engineering* (2017): 195-216.
[165] Larrucea, Xabier, et al. "Microservices." *IEEE Software* 35.3 (2018): 96-100.
[166] Stocker, Mirko, et al. "Interface quality patterns: Communicating and improving the quality of microservices Apis." *Proceedings of the 23rd European Conference on Pattern Languages of Programs*. 2018.
[167] Laigner, Rodrigo, et al. "Data management in microservices: State of the practice, challenges, and research directions." *arXiv preprint arXiv:2103.00170* (2021).
[168] Hemon, Aymeric, et al. "From agile to DevOps: Smart skills and collaborations." *Information Systems Frontiers* 22.4 (2020): 927-945.
[169] Alata, Eric, et al. "Lessons learned from the deployment of a high-interaction honeypot." *2006 Sixth European Dependable Computing Conference*. IEEE, 2006.
[170] Guo, Chuanxiong, et al. "Dcell: a scalable and fault-tolerant network structure for data centers." *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*. 2008.
[171] Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." *Proceedings of the IEEE* 103.1 (2014): 14-76.
[172] Warrender, Christina, Stephanie Forrest, and Barak Pearlmutter. "Detecting intrusions using system calls: Alternative data models." *Proceedings of the 1999 IEEE symposium on security and privacy (Cat. No. 99CB36344)*. IEEE, 1999.
[173] Forrest, Stephanie, et al. "A sense of self for unix processes." *Proceedings 1996 IEEE Symposium on Security and Privacy*. IEEE, 1996.
[174] Abed, Amr S., T. Charles Clancy, and David S. Levy. "Applying bag of system calls for anomalous behavior detection of applications in linux containers." *2015 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2015.
[175] Tunde-Onadele, Olufogorehan, et al. "A study on container vulnerability exploit detection." *2019 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE, 2019.
[176] Pfaff, Ben, et al. "The Design and Implementation of Open {vSwitch}." *12th USENIX symposium on networked systems design and implementation (NSDI 15)*. 2015.
[177] Mairh, Abhishek, et al. "Honeypot in network security: a survey." *Proceedings of the 2011 international conference on communication, computing & security*. 2011.
[178] Sokol, Pavol, Matej Zuzčák, and Tomáš Sochor. "Definition of attack in the context of low-level interaction server honeypots." *Computer Science and its Applications*. Springer, Berlin, Heidelberg, 2015. 499-504.
[179] Chovancová, Eva, and Norbert Ádám. "The Security of Heterogeneous Systems based on Cluster High-interaction Hybrid Honeypot." *2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES)*. IEEE, 2019.
[180] Wang, He, and Bin Wu. "SDN-based hybrid honeypot for attack capture." 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, 2019.
[181] Osman, Amr, et al. "Sandnet: towards high quality of deception in container-based microservice architectures." *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019.
[182] Kyriakou, Andronikos, and Nicolas Sklavos. "Container-based honeypot deployment for the analysis of malicious activity." *2018 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, 2018.
[183] Young, Ethan G., et al. "The True Cost of Containing: A {gVisor} Case Study." *11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19)*. 2019.
[184] Kumar, Rakesh, and B. Thangaraju. "Performance analysis between runc and kata container runtime." *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*. IEEE, 2020.
[185] Sever, Dubravko, and Tonimir Kišasondi. "Efficiency and security of docker based honeypot systems." *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2018.
[186] Viktorsson, William, Cristian Klein, and Johan Tordsson. "Security-performance trade-offs of kubernetes container runtimes." *2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE, 2020.

[187] Combe, Theo, Antony Martin, and Roberto Di Pietro. "To docker or not to docker: A security perspective." *IEEE Cloud Computing* 3.5 (2016): 54-62.
[188] Reti, Daniel, and Norman Becker. "Escape the Fake: Introducing Simulated Container-Escapes for Honeypots." *arXiv preprint arXiv:2104.03651* (2021).
[189] Gomes, Jorge, et al. "Enabling rootless Linux Containers in multi-user environments: the udocker tool." *Computer Physics Communications* 232 (2018): 84-97.
[190] Tinney, Macdara. "Intrusion Detection for Kubernetes Based Cloud Deployments." (2020).
[191] Tsikerdekis, Michail, et al. "Approaches for preventing honeypot detection and compromise." *2018 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, 2018.
[192] Gupta, Chakshu. *HoneyKube: designing a honeypot using microservices-based architecture*. MS thesis. University of Twente, 2021.
[193] Bontaş, Carol-Sebastian, Ioan-Mihail Stan and Răzvan Rughiniş. “Honeypot generator using software defined networks and recursively defined topologies.” *2022 21st RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, 2022