Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Ingineria Software și Automatică

**RAPORT**

Lucrare de laborator Nr.7

Disciplina: Analiza și Modelarea Sistemelor

Tema: Dezvoltarea elaborărilor prin intermediul diagramelor de componente şi de plasare

A efectuat: st. Vremere Adrian

gr. FAF-232

A verificat : asist.univ.   
 Sava Nina

lect.univ.  
Melnic Radu

Chișinău 2024

**Scopul**: Studierea noțiunilor de component, interfața, nod, dependență, conexiune între noduri. **Sarcina:** De realizat 5 diagrame de componente și 1 diagrama de plasare pentru sistemul de operare MacOS.

**Considerații teoretice:**

Diagrama componentelor, spre deosebire de diagramele cercetate, descrie particularitățile reprezentării fizice ale unui sistem. Diagrama componentelor permite determinarea arhitecturii sistemului dezvoltat prin stabilirea dependenței dintre componentele programului, care pot fi codul sursă inițial, codul binar și cel executabil. În mai multe domenii de dezvoltare, modulul și componenta corespund fișierului. Săgețile punctate care leagă modulele indică relațiile de dependență, asemănătoare celor care apar la compilarea codurilor sursă inițiale. Elementul grafic de bază al diagramei componentelor sunt componentele, interfețele și dependențele dintre acestea.

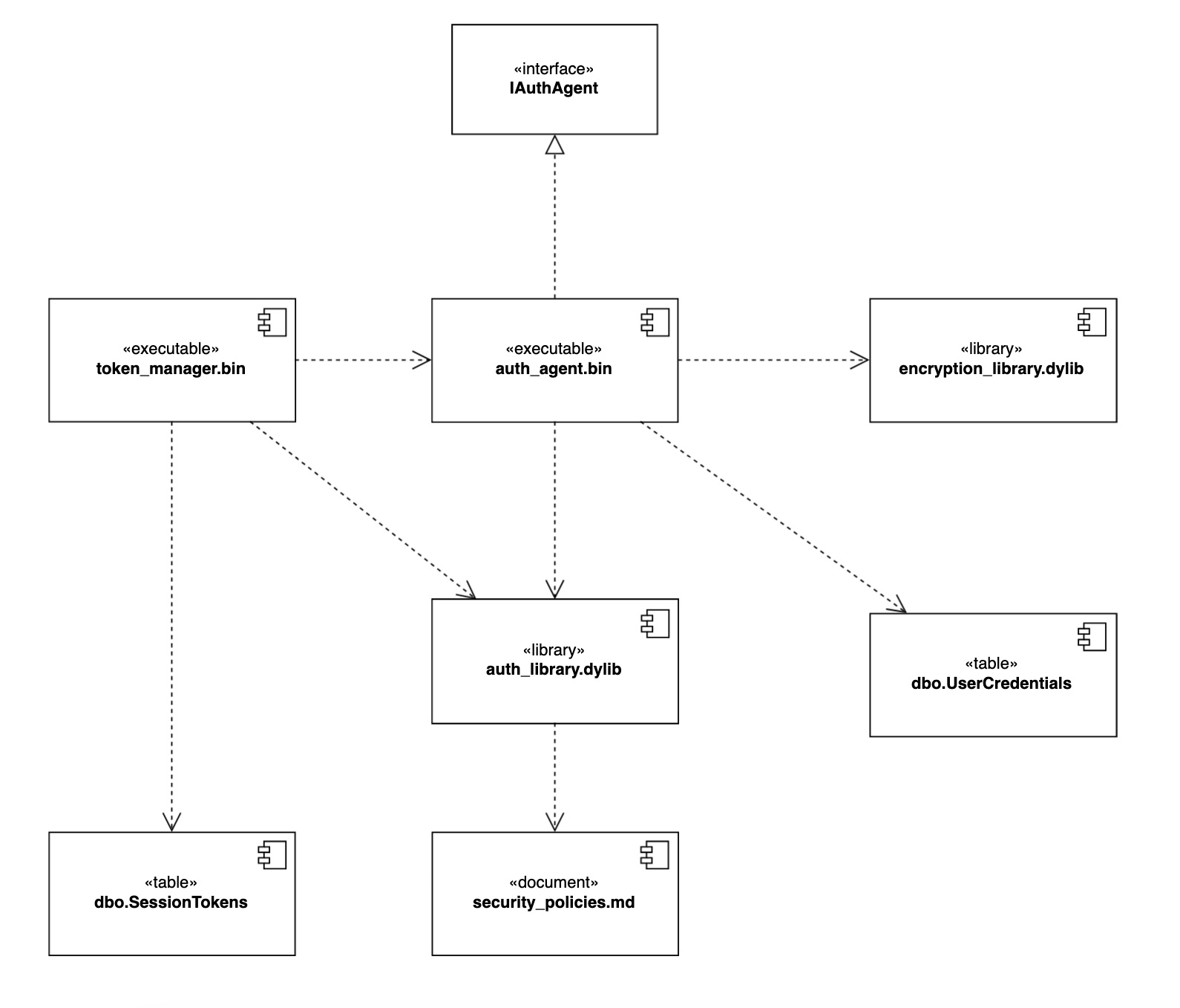
Diagrama componentelor se dezvoltă în următoarele scopuri: • Vizualizarea structurii comune a codului sursă al unui sistem de program. • Specificarea versiunii executabile a unui sistem de program. • Asigurarea reutilizării repetate a unor fragmente din codul sursă. • Reprezentarea conceptuală și fizică a schemelor bazei de date.

A doua formă de reprezentare fizică a sistemului de programare este diagrama de plasare. Aceasta este utilizată pentru reprezentarea generală a configurației și topologiei sistemului de programare distribuit și conține distribuirea componentelor pe nodurile sistemului. Pe lângă indicarea diagramei de plasare, aceasta mai arată și conexiunile fizice – rutele de transfer al informațiilor între echipamentele utilizate în implementarea sistemului.

Diagrama de plasare este specifică pentru vizualizarea elementelor și componentelor programului, care există doar în stadiul execuției acestuia (runtime). Drept urmare, doar componentele – copii ale programului, care sunt fișiere executabile sau biblioteci dinamice, sunt prezentate. Componentele care nu sunt utilizate în stadiul execuției nu sunt indicate în diagrama de plasare. Componentele cu textele programului inițial pot fi prezentate doar în diagrama componentelor. Ele nu sunt indicate în diagrama de rezervare.

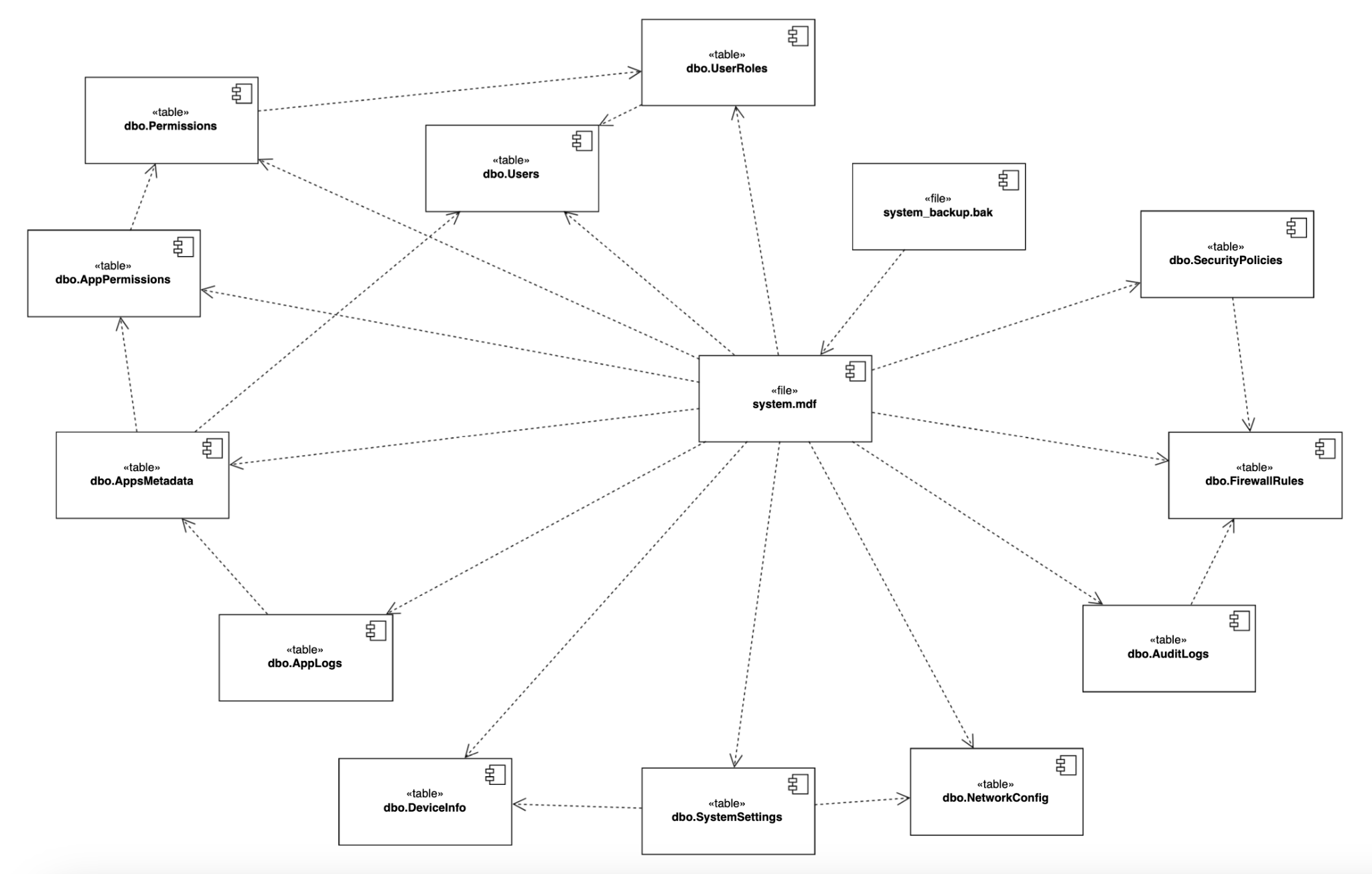
MacOS este sistemul de operare proprietar al Apple pentru computerele Macintosh, renumit pentru designul său elegant și interfața intuitivă. Lansat în 2001 sub denumirea „Mac OS X”, acesta a evoluat într-un sistem modern și robust care oferă integrare perfectă cu alte dispozitive Apple, precum iPhone-uri și iPad-uri. Bazat pe o arhitectură Unix, macOS este recunoscut pentru stabilitatea și eficiența sa.

**Implementare, rezultate practice:**



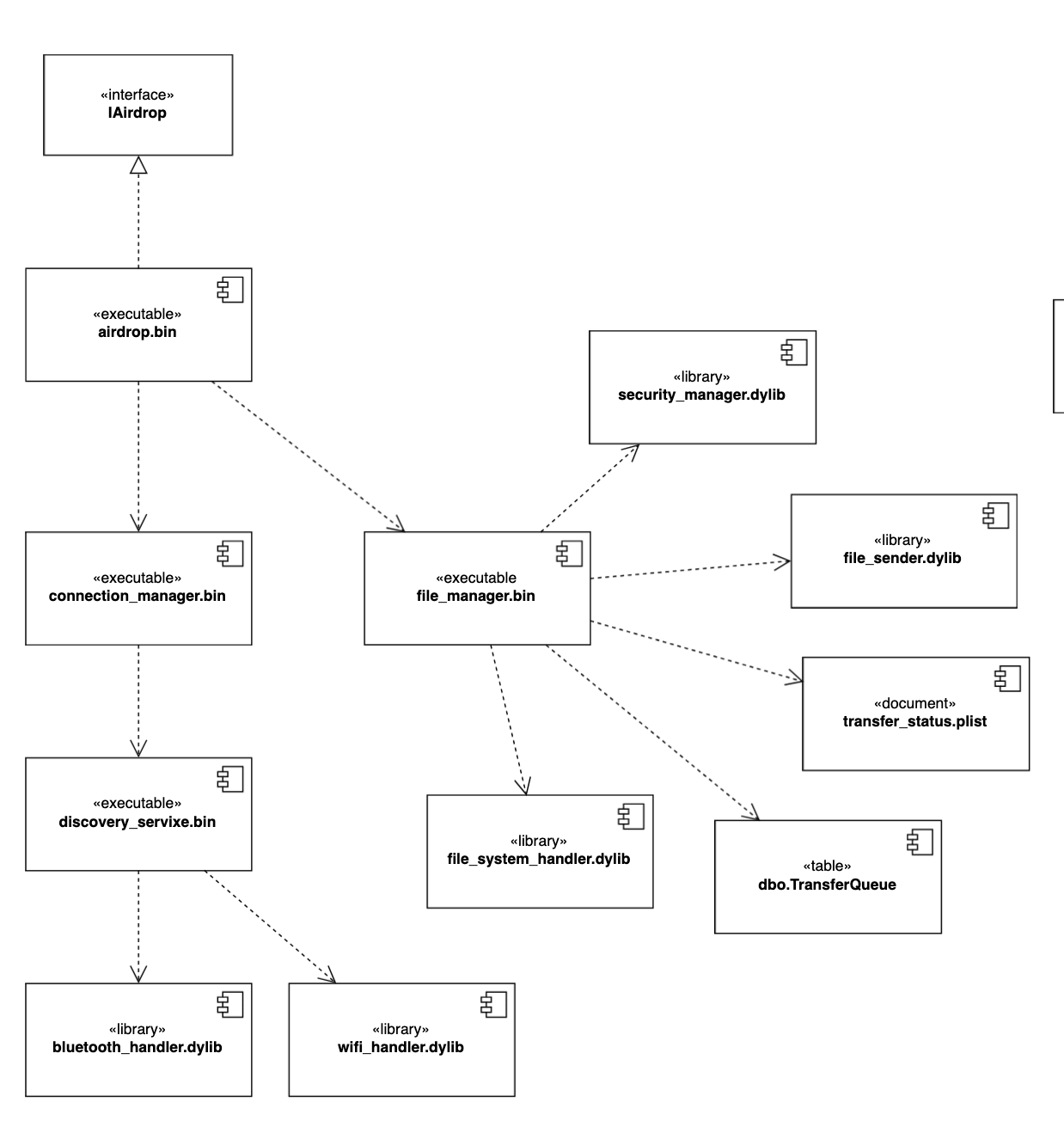
***Figura 1.*** Diagrama de componente – Modulul de creare a unei sesiuni în sistemul MacOS

Diagrama din **figura 1** descrie comportamentul unui sistem de autentificare, evidențiind interacțiunile dintre componentele sale principale. Interfața IAuthAgent definește contractul de autentificare, implementat de executabilul *auth\_agent.bin*, care comunică atât cu librăria de criptare (*encryption\_library.dylib*), cât și cu librăria de autentificare (*auth\_library.dylib*). Executabilul *token\_manager.bin* gestionează token-urile de sesiune, interacționând cu baza de date *dbo.SessionTokens*. De asemenea, sistemul stochează datele utilizatorilor în tabelul *dbo.UserCredentials* și include un document de referință pentru politicile de securitate (*security\_policies.md*). Toate componentele colaborează pentru a asigura securitatea și gestionarea eficientă a sesiunilor.

******

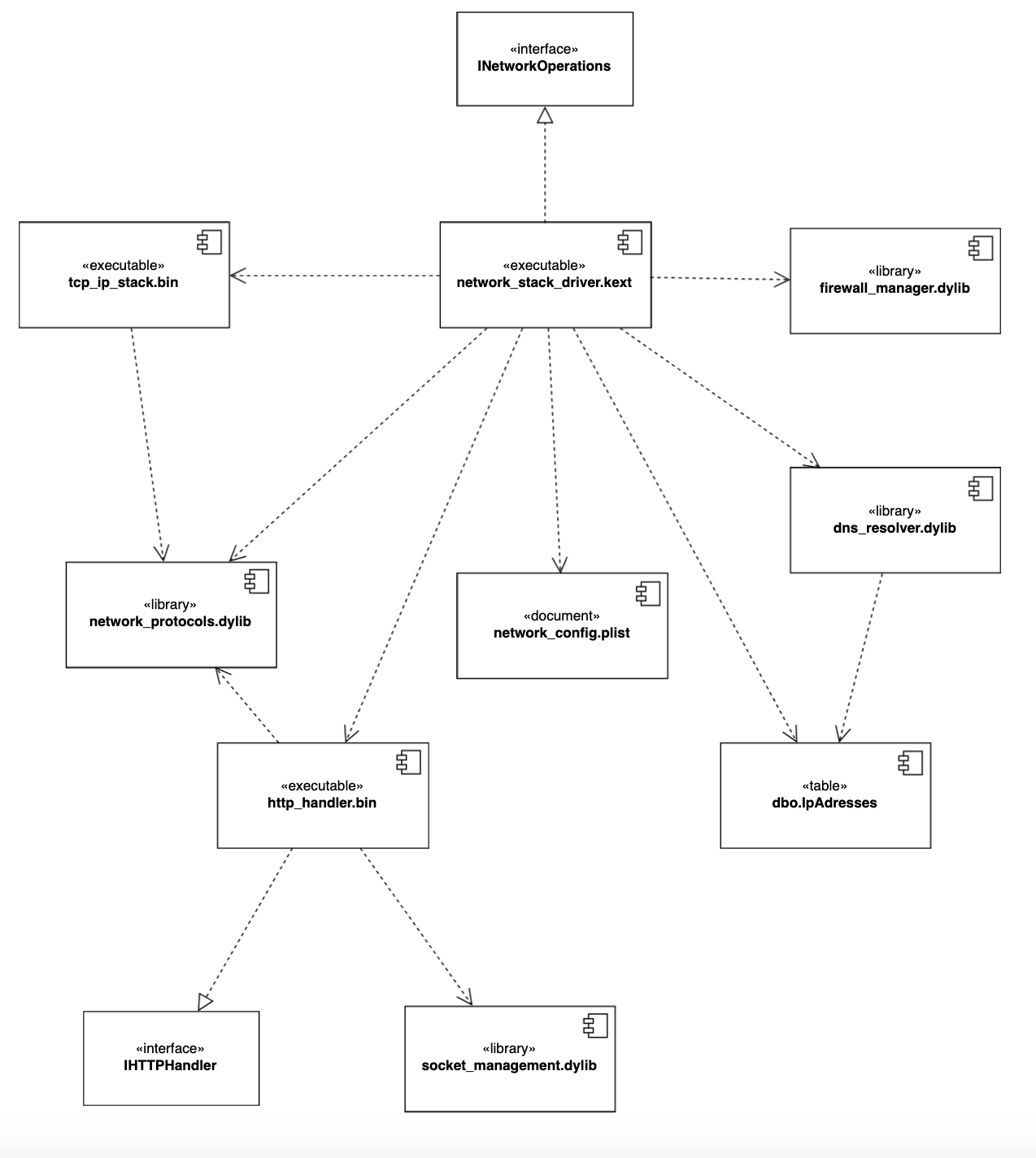
***Figura 2.*** Diagramă de componente – Baza de date a sistemului MacOS

Această diagramă din **figura 2** reprezintă structura și relațiile dintre diferite componente ale unei baze de date centralizate, reprezentată de fișierul principal system.mdf. Tabelele stochează informații esențiale despre utilizatori (*dbo.Users*), rolurile acestora (*dbo.UserRoles*), permisiuni (*dbo.Permissions* și *dbo.AppPermissions*), metadatele aplicațiilor (*dbo.AppsMetadata*), și logurile acestora (*dbo.AppLogs*). De asemenea, sunt incluse tabele pentru configurarea dispozitivelor (*dbo.DeviceInfo*), a rețelei (*dbo.NetworkConfig*), a setărilor de sistem (*dbo.SystemSettings*), și a regulilor de securitate (dbo.FirewallRules). Sistemul menține un istoric al politicilor de securitate (*dbo.SecurityPolicies*) și al activităților prin tabele de audit (*dbo.AuditLogs*). Copia de rezervă este reprezentată de fișierul *system\_backup.bak*, asigurând protecția datelor în caz de pierdere. Toate componentele sunt conectate prin relații logice pentru a sprijini administrarea eficientă și securizată a datelor.



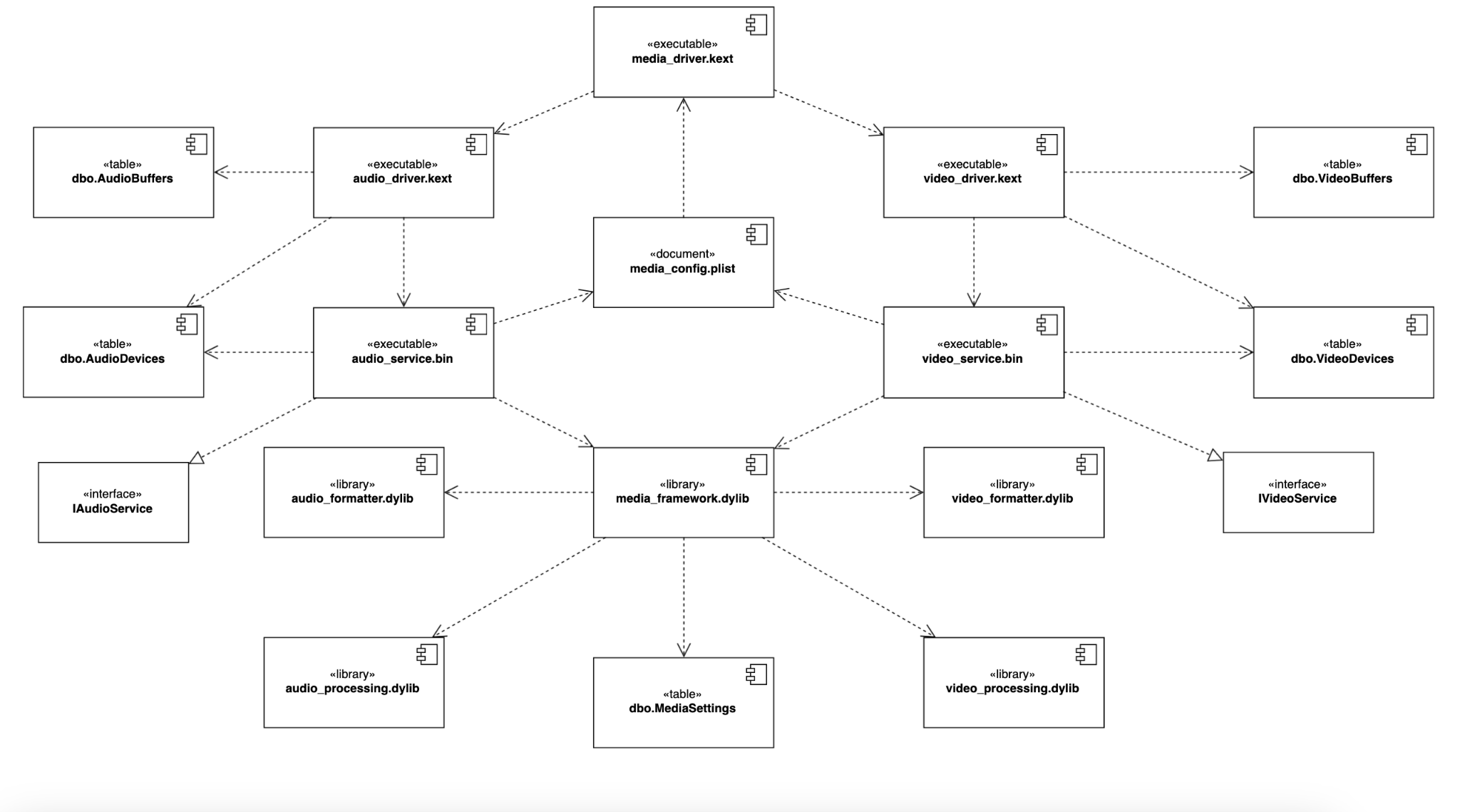
***Figura 3.*** Diagramă de componente – Modulul de transferare a fișierelor în sistemul MacOS

Această diagrama din **figura 3** ilustrează arhitectura unui sistem de transfer de fișiere, bazat pe interfața *IAirdrop*, implementată de executabilul principal airdrop.bin. Acesta colaborează cu componentele *connection\_manager.bin* și file\_manager.bin pentru gestionarea conexiunilor și manipularea fișierelor. *connection\_manager.bin* utilizează serviciul de descoperire (*discovery\_service.bin*), care interacționează cu librăriile pentru gestionarea conexiunilor Bluetooth (*bluetooth\_handler.dylib*) și Wi-Fi (*wifi\_handler.dylib*). file\_manager.bin integrează librării precum *security\_manager.dylib* pentru securitate, *file\_sender.dylib* pentru trimiterea fișierelor și *file\_system\_handler.dylib* pentru gestionarea sistemului de fișiere. Starea transferurilor este documentată în t*ransfer\_status.plist*, iar coada transferurilor este stocată în tabelul *dbo.TransferQueue*. Sistemul este structurat pentru a asigura transferuri rapide și sigure prin conexiuni multiple.



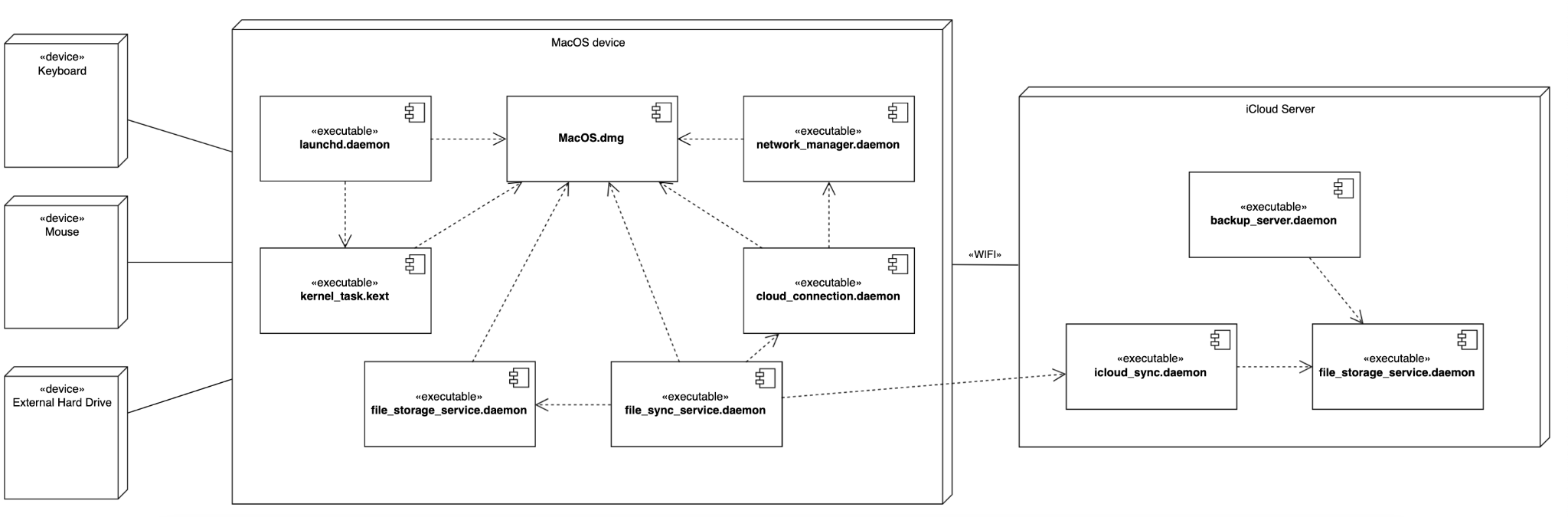
***Figura 4.*** Diagramă de componente – Modulul de conexiuni în sistemul MacOS

Această diagramă de componente din **figura 4** reprezintă modulul de gestionare a rețelei în sistemul MacOS, având la bază un modul central numit *network\_stack\_driver.kext*, care implementează interfața *INetworkOperations*. Modulul central interacționează cu mai multe componente, inclusiv biblioteci (*firewall\_manager.dylib*, *dns\_resolver.dylib*, *network\_protocols.dylib*, și *socket\_management.dylib*), un document de configurare (*network\_config.plist*), și o bază de date (*dbo.IpAdresses*). De asemenea, acesta coordonează execuția altor module precum *tcp\_ip\_stack.bin* și *http\_handler.bin*, acesta din urmă interacționând cu o interfață specifică, *IHTTPHandler*, și cu biblioteca de gestionare a conexiunilor de rețea, *socket\_management.dylib*. Structura ilustrează relațiile de dependență dintre modulele software, evidențiind utilizarea fișierelor de configurare, a bibliotecilor partajate și a interfețelor pentru gestionarea rețelei.

******

***Figura 5.*** Diagramă de componente – Modulul de audio și video al sistemului MacOS

Această diagramă din **figura 5** ilustrează componentele unui sistem media din macOS, bazat pe un modul central, *media\_driver.kext*, care gestionează servicii audio și video prin interacțiuni cu module și resurse specifice. Modulul *audio\_driver.kext* și *video\_driver.kext* sunt responsabile pentru manipularea hardware-ului audio și video, interacționând cu tabele de date precum *dbo.AudioBuffers*, *dbo.AudioDevices*, *dbo.VideoBuffers* și *dbo.VideoDevices*. Acestea se conectează la servicii precum audio\_service.bin și video\_service.bin, care implementează interfețele *IAudioService* și *IVideoService*. Configurațiile centrale sunt gestionate prin documentul *media\_config.plist*. Biblioteci partajate, inclusiv *media\_framework.dylib*, a*udio\_formatter.dylib*, *audio\_processing.dylib*, *video\_formatter.dylib*, și *video\_processing.dylib*, furnizează funcționalități suplimentare pentru procesare și formatare. În centrul arhitecturii se află *media\_framework.dylib*, care integrează și conectează toate componentele pentru a susține operațiunile media complexe.

******

***Figura 6.*** Diagramă de plasare – Sistemul MacOS

Această diagramă de plasare din **figura 6** ilustrează interacțiunile dintre componentele hardware, software și serviciile de cloud într-un sistem macOS. Dispozitivele fizice, cum ar fi tastatura, mouse-ul și un hard disk extern, sunt conectate la un dispozitiv macOS care rulează sistemul de operare reprezentat de MacOS.dmg. Acesta gestionează diverse procese prin intermediul unor daemoni precum launchd.daemon, care inițiază componente esențiale, inclusiv kernel\_task.kext pentru gestionarea resurselor de bază și network\_manager.daemon pentru conectivitate în rețea. Pentru funcționalități de stocare și sincronizare, daemoni precum *file\_storage\_service.daemon* și *file\_sync\_service.daemon* interacționează atât local, cât și cu serviciile cloud. Conectivitatea Wi-Fi permite comunicația cu serverul iCloud, unde *icloud\_sync.daemon* și *backup\_server.daemon* asigură sincronizarea și backup-ul datelor. Această arhitectură subliniază integrarea dintre dispozitive fizice, procese software locale și servicii cloud într-un mediu macOS.

**Concluzii:**

În concluzie, după finalizarea acestui laborator dedicat studiului noțiunilor de componentă, interfață, nod, dependență și conexiune între noduri, precum și crearea diagramelor de componente și a diagramei de plasare pentru sistemul informatic ales – în cazul meu, MacOS – am dobândit perspective valoroase în ceea ce privește realizarea acestor diagrame. Conform cerințelor, m-am asigurat să creez și să reprezint cinci diagrame de componente ale sistemului, inclusiv cele pentru modulele de Audio și Video, Conexiuni, Logare și alte module, utilizând elementele necesare, precum componentele și interfețele, respectând o convenție de denumire corespunzătoare și relațiile corecte între elemente, cum ar fi dependențele dintre componente și realizările între componente și interfețe. În plus, am realizat diagrama de plasare a sistemului informatic ales într-un mod corect, folosind noduri, dispozitive, componente, interfețe și toate elementele cerute, precum și relațiile de asociere între noduri sau dispozitive și dependențele dintre componente. Această experiență m-a ajutat să vizualizez sistemul MacOS într-un mod structurat și clar, la fel cum au făcut-o toate lucrările de laborator pe care le-am finalizat anterior.

**Bibliografie**

1. **Melnic R., Sava N.** Îndrumar metodic “Analiza si modelarea sistemelor informationale”
2. **Sava N.** Îndrumar metodic “Regulile sintaxei - Diagrama componentelor”
3. **Sava N.** Îndrumar metodic “Regulile sintaxei - Diagrama desfășurărilor”
4. **MacOS | Wikipedia** [Sursă Electronică] Mod de Acces:

<https://en.wikipedia.org/wiki/MacOS>