

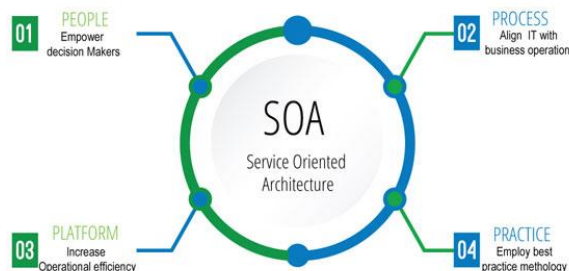
## Service Oriented

### Arhitectura orientate pe servicii:

- Modalitate de dezvoltare a sistemelor distribuite în care componentele sunt servicii stand-alone.
- Serviciile se pot executa pe diverse stații ale unor provideri de servicii diferiți.
- Protocoale standardizate Protocoale standardizate pentru suportul pentru suportul comunicației între servicii și schimbului de informații.

### Caracteristici:

- Interoperabilitate între diferite platforme și limbaje de programare.
- Blocurile funcționale sunt încapsulate în componente care funcționează ca servicii.
- Separarea interfețelor de implementare.
- În cazul aplicațiilor complexe, controlul fluxului de execuție (workflow) este separat de servicii.
- Serviciile pot fi adăugate dinamic.
- Corespondența dintre interfețe și implementări se realizează prin fișiere de configurare și poate fi adaptată.



**SOA** implică implementarea de servicii, acestea sunt unități de logică care rulează într-o rețea. Un serviciu are următoarele caracteristici: acesta gestionează un proces de afaceri, cum ar fi calcularea unei cote de asigurare sau distribuirea unui e-mail; gestionează o sarcină tehnică, cum ar fi accesarea unei baze de date; sau oferă date de afaceri și detalii tehnice pentru a construi o interfață grafică.

Poate accesa un alt serviciu, cu tehnologia de rulare corespunzătoare, poate accesa un program tradițional și poate răspunde la diferite tipuri de solicitanți, cum ar fi aplicațiile web. Este relativ independent de alte programe. Modificările unui solicitant necesită puține modificări efectuate în cadrul serviciului. Modificările la logica internă a unui serviciu necesită puține modificări sau nu sunt necesare pentru solicitant. Independența relativă a serviciului și a altor programe se numește cuplaj liber.

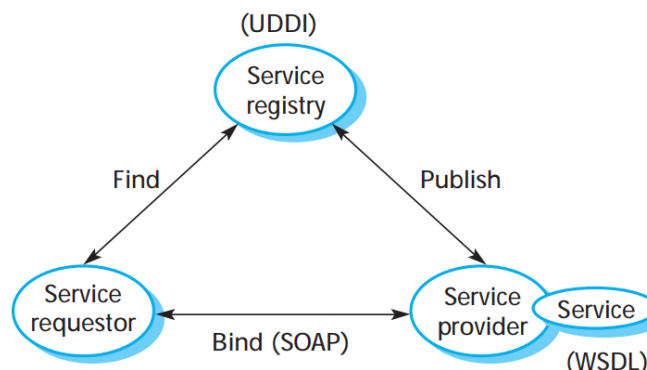
**SOA** implică un stil de dezvoltare care se concentrează asupra modularității și reutilizării și nu este doar pentru un cod nou. Este posibil să migrezi aplicațiile existente în următoarele cazuri:

- Aplicațiile monolitice, care combina logica interfeței cu utilizatorul și accesul la date, astfel încât actualizarea unui tip de logică necesită asistența pentru testarea mai multor tipuri de comportament.
- Aplicațiile care sunt greu de înțeles deoarece logica lor este monolită și au fost făcute patch-uri în mod repetat, în loc să fie rescris codul. Actualizările au nevoie de timp în plus, deoarece programatorii încearcă să descifreze logica și pe măsură ce complexitatea crește, erorile suplimentare însoțesc actualizările.
- Inventarul aplicației are o logică dublă. Cererile de schimbare sunt perturbatoare, necesitând schimbări în mai multe locuri.

*De ce SOA*, realitatea este că în întreprinderile IT infrastructura este eterogenă între sistemele de operare, aplicații, software-ul și infrastructura aplicațiilor, unele aplicații existente sunt utilizate pentru a rula procesele de afaceri curente, așa că pornirea de la zero pentru a construi o nouă infrastructură nu este o opțiune.

Companiile ar trebui să răspundă repede la schimbările de afaceri, acestea ar trebui să stimuleze investițiile în aplicații și infrastructuri de aplicații pentru a răspunde ulterior cerințelor de afaceri mai noi, să sprijine noi canale de interacțiune cu clienții, partenerii și furnizorii; și oferă o arhitectură care susține afacerea ecologică.

SOA, cu natura sa cuplată în mod deliberat, permite întreprinderilor să conecteze noi servicii sau să actualizeze serviciile existente într-o manieră granulară pentru a răspunde noilor cerințe de afaceri, oferă opțiunea de a face serviciile consumabile pe diferite canale și expune aplicațiile existente și aplicațiile moștenite ca servicii, protejând astfel investițiile existente în infrastructura IT.



### Avantaje SOA:

- Serviciile pot fi furnizate local sau pot fi furnizate de către provideri externi.
- Serviciile pot fi independente de limbaj.
- Investiția în sistemele legacy poate fi conservată.
- Interoperabilitatea este facilitată prin schimbul simplificat de informații.