

[P271]

9

Информациони системи



Саша Малков
Универзитет у Београду
Математички факултет
2023/2024

[P271]

Информациони системи

Саша Малков



Тема 13

Архитектуре оријентисане према сервисима – COA

[P271] Информациони системи – Саша Малков – 2023/24 – час 9

1

Софтверске архитектуре

Улога софтверске архитектуре



- Основни задатак софтверске архитектуре је да пружи
 - решење проблема
 - перформансе
 - флексибилност
 - проширивост
- Чест проблем је задржавање на прва два аспекта, без довољног разматрања друга два

Универзитет у Београду - Математички факултет

[P271] Информациони системи – Саша Малков – 2023/24 – час 9

2

Софтверске архитектуре

Флексибилност архитектуре



- Флексибилност архитектуре представља способност
 - да се читава архитектура
 - прилагођава променама у окружењу
 - примењује у новим околностима
 - да се елементи архитектуре
 - прилагођавају променама у окружењу
 - примењују у новим околностима
 - примењују у другим новим пројектима
- Флексибилност је блиска поновној употребљивости

Универзитет у Београду - Математички факултет

[P271] Информациони системи – Саша Малков – 2023/24 – час 9

3



Проширивост архитектуре

- Проширивост архитектуре је својство архитектуре
 - да може да се проширује
 - да се проширивање остварује
 - примарно додавањем нових елемената
 - без значајног мењања постојећих елемената



Флексибилност и проширивост

- омогућавају поједностављивање животног циклуса система за све учеснике
 - модуларност
 - конфигурабилност
 - робусност
 - могућност брзе реакције на промене у пословним захтевима
 - и друго



Сервисно-оријентисана архитектура

- (*Service-Oriented Architecture, SOA*)
- Резултат еволуције софтверске индустрије са циљем постизања максималне флексибилности и проширивости
- Модуларизацијом се целине система обликују у компоненте које могу да “самостално” пружају одређене “целовите” услуге
- Такве компоненте се називају *сервиси*



Дефиниција

- СОА је архитектурални стил који промовише коришћење *лабаво повезаних сервиса* ради обезбеђивања *високе флексибилности* на *инијеројерабилан* и технолошки независан начин...



Уопштавање комуникације

- Да би сервиси били поновно примењиви, потребно је да имају релативно стандардизоване начине повезивања и остваривања комуникације



Уопштавање комуникације (2)

- На нивоу архитектуре СОА неопходно је да
 - логика буде раздвојена од комуникације
 - да сви сервиси једног система користе исте протоколе комуникације и повезивања
 - да протоколе комуникације и повезивања имплементира посебна компонента



Уопштавање комуникације (3)

- На нивоу архитектуре (а не технологије) се постиже висок ниво уопштавања и флексибилности
 - компонента за повезивање обухвата сав технолошки зависан код
 - остатак кода је лако преносив
 - у случају промене начина комуникације, мења се или замењује “само” компонента за комуникацију
 - уз минималне измене осталих компоненти



Појам апликације

- СОА је архитектурална парадигма која се примарно односи на дизајн софтвера (а не на технологију)
- Примена концепта СОА води промени концепта апликације
- **Апликација представља композицију и груписање лабаво повезаних, уско дефинисаних и на стандардима утемељених сервиса у веће и сложеније целине**

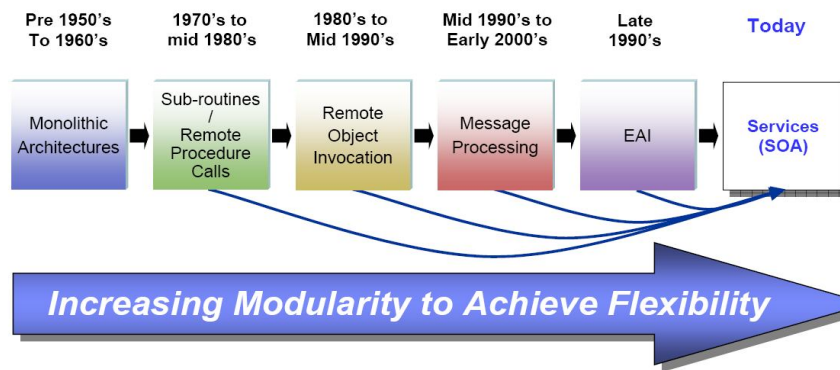


Мотивација

- Савремени пословни процеси су динамични и дистрибуирани
- одликују се јасном поделом одговорности међу учесницима и деловима процеса
- флукутирају, при чему су промене најчешће локалног карактера – односе се на поједине делове или учеснике
- Софтверска имплементација таквих процеса мора да прати њихову логику
 - SOA то врло добро чини



Историјат архитектура



Историјат архитектура (2)

- **Монолитна архитектура**
 - све (подаци, кориснички интерфејс, обрада) се налази у оквиру једног система
- **Потпрограми и удаљене процедуре**
 - клијент ("главни" програм) шаље захтев познатом извршиоцу (потпрограму, удаљеном серверу) са датим параметрима
 - клијент добија одговор и наставља са радом
- **Позивање удаљених објеката**
 - концепт пословних објеката – делова кода који укључују податке и понашање који могу да се мењају у зависности од контекста
 - системи дистрибуираних објеката
- ...



Историјат архитектура (3)

- ...
- **Обрада порука**
 - архитектуре дистрибуираних објеката су углавном јако спрегнуте
 - архитектура обраде порука апстрахује интеракцију концептом размене порука ради спуштања нивоа спрегнутости
 - овакве поруке имају већу грануларност него поруке међу објектима
 - старање о трансакцијама се (бар делимично) преноси на ниво порука
- **Интеграција пословних апликација**
 - (*Enterprise Application Integration, EAI*)
 - унапређивање и спајање претходних архитектура
 - удаљени објекти: CORBA, COM,...
 - редови порука
 - размена података преко XML-а
 - вишеслојне архитектуре и друго
 - различити делови апликације могу да користе различите архитектуре
- COA

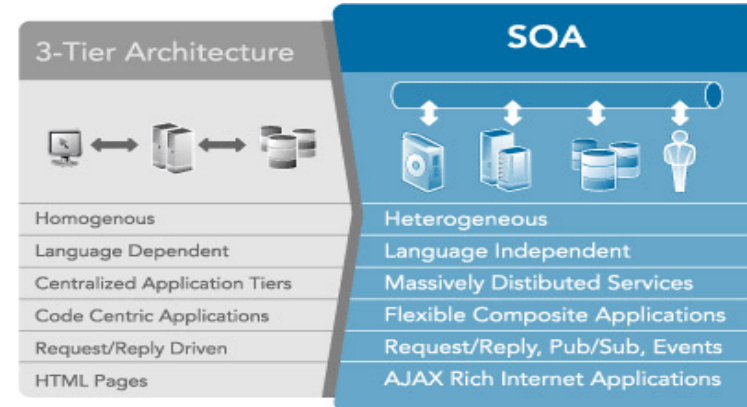


Одлике СОА

- Софтверски елементи као “самостални” сервиси
- Отворени стандарди повезивања
- Пројекат архитектуре не улази у детаље имплементације већ се бави “састављањем”
- Објектно-оријентисане методологије развоја
- Савремени принципи развоја:
 - модуларност
 - енкапсулација
 - ниска спрегнутост
 - поновна употребљивост
 - имплементација као састављање
 - ...



СОА и вишеслојне архитектуре



СОА и дистрибуирани објекти

- Системи за дистрибуирање објеката (*CORBA, COM,...*) као основне јединице имају ентитете
- СОА као основне јединице има *услуге (сервисе)*
- Сервис одговара једном заокруженом елементу пословања
 - као аутоматизован или слабо интерактиван *случај употребе*



СОА из различитих угла

- Из пословног угла
 - СОА је скуп пословних сервиса
- Из угла софтверске архитектуре
 - Сервиси су делови сложене архитектуре коју чине
 - пружалац сервиса (*service provider*) и
 - корисник сервиса (*consumer*) и
 - опис интерфејса сервиса (*interface-based service description*)
- Из угла имплементације
 - Сервиси се имплементирају применом отворених стандарда и протокола као што су, на пример, Веб сервиси



Сервиси

- Сервис је
 - издвојена (дискретна) јединица пословне (техничке) функције
 - која је поновно употребљива



Сервиси (2)

- Сервиси су:
 - повезиви
 - слабо спрегнути
 - дефинисани су интерфејсима независним од имплементације
 - корисник сервиса зависи само до интерфејса
 - промовише се флексибилност при изменама имплементације
 - транспарентни у односу на локацију
 - не користе друге сервисе на основу локације
 - међусобно сарађују (интероперабилни)
 - расположиви независно од имплементације или протокола преноса
 - корисници и пружаоци сервиса могу бити имплементирани на различитим платформама
 - комуникациони протоколи сервиса треба да буду компатибилни са различитим платформама
 - отворени стандарди



Сервиси (3)

- Нуди их пружаоц сервиса (*Service Provider*)
 - објављују интерфејсе и политике
- Користе их корисници сервиса (*Service Consumer*)
 - користе интерфејсе
 - примена поузданих и безбедних протокола
- Посредују медијатори сервиса (*Service mediators*)
 - обезбеђују откривање, избор, праћење рада и друго



Описивање сервиса

- Сервиси се описују применом стандардизованих поступака за описивање интерфејса
 - тзв. *меташајдога*ци сервиса
 - одређују се сервис, његове операције, улазне и излазне параметре
 - одређује се како се сервису приступа, као и која је његова локација (*service endpoint*)



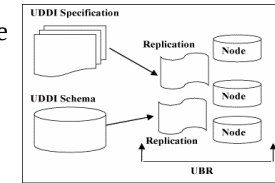
Регистар сервиса

- Представља вид *каталога* путем кога могу да се објављују (*publish*) и проналазе (*discover*) сервиси
- Сервиси објављују организације које пружају сервисе
 - радне јединице у организацији
 - организације
- Регистар сервиса је средство за додатно спуштање нивоа спрегнутости
 - корисник сервиса не мора да зна
 - ко пружа сервис
 - где се физички налази имплементација сервиса
 - у неким случајевима чак ни све детаље интерфејса (операција)



Universal Discovery, Description, Integration (UDDI)

- УДДИ
 - окружење које је независно од платформе
 - служи за
 - описивање сервиса
 - отварање пословања
 - интегрисање пословних сервиса
- Чине га
 - модел података – XML схема која описује сервисе
 - каталог са информацијама о Веб сервисима
 - подршка за откривање сервиса
 - спецификације АПИ за претраживање и објављивање података
 - средства за успостављање односа између пружаоца и корисника сервиса



Врсте сервиса

- Основне врсте сервиса су:
 - Сервиси пословних процеса
 - Сервиси пословних трансакција
 - Сервиси пословних функција
 - Сервиси техничких функција



Врсте сервиса (2)

- Основне врсте сервиса су:
 - Сервиси пословних процеса
 - (*business process services*)
 - Пословни процеси су секвенце активности које испуњавају један пословни циљ. Један процес се може представити као сервис
 - На пример:
 - процес пријаве за добијање кредита
 - Сервиси пословних процеса се могу састојати од других сервиса (тј.могу их користити)
 - Сервиси пословних трансакција
 - Сервиси пословних функција
 - Сервиси техничких функција



Врсте сервиса (3)

- Основне врсте сервиса су:
 - Сервиси пословних процеса
 - **Сервиси пословних трансакција**
 - (*business transaction services*)
 - Представљају пословне функције које мењају неко стање пословања
 - На пример:
 - набавка материјала
 - пренос средстава
 - свако мењање пословних података
 - Сервиси пословних функција
 - Сервиси техничких функција



Врсте сервиса (4)

- Основне врсте сервиса су:
 - Сервиси пословних процеса
 - Сервиси пословних трансакција
 - **Сервиси пословних функција**
 - (*business function services*)
 - Сервиси пословних функција представљају пословне функције који не мењају стање пословних процеса
 - или испоручују податке
 - или извршавају једноставнија израчунавања
 - На пример:
 - конверзија валута
 - извештаји (у форми података)
 - Сервиси техничких функција



Врсте сервиса (5)

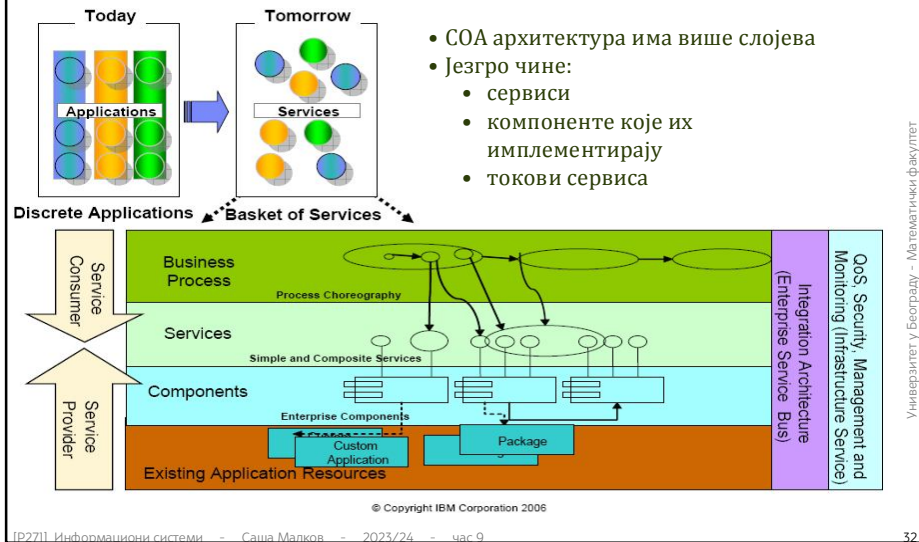
- Основне врсте сервиса су:
 - Сервиси пословних процеса
 - Сервиси пословних трансакција
 - Сервиси пословних функција
 - **Сервиси техничких функција**
 - (*technical function services*)
 - Поновно употребљиви сервиси који не пружају пословне функције, већ обезбеђују техничке или инфраструктурне функције
 - На пример:
 - праћење догађаја
 - провера ауторизација
 - провера стања система, расположивости сервиса и сл.



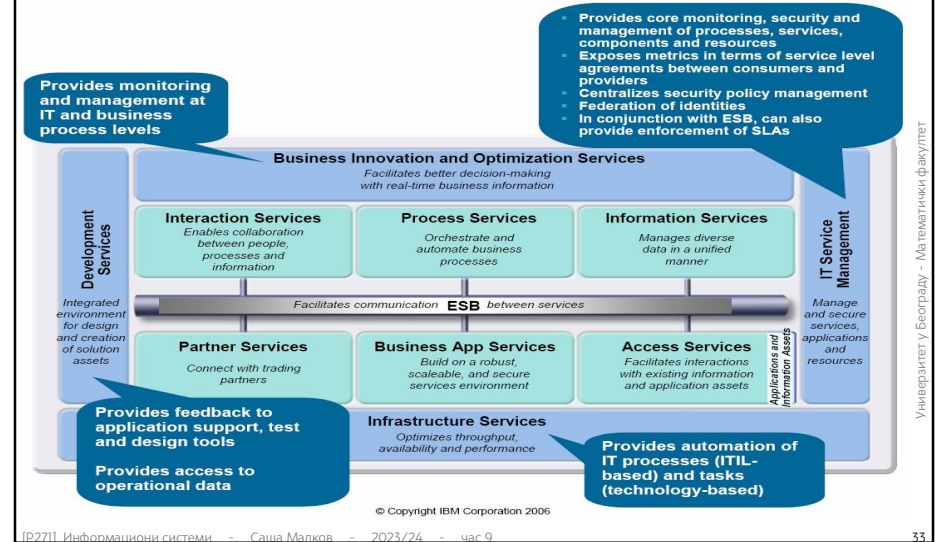
Сервис као фасада

- За постојећу апликацију се могу направити и њој доделити сервиси “*facade*”
 - кроз интерфејс сервиса излажу неке функције апликације
 - сакривају имплементацију

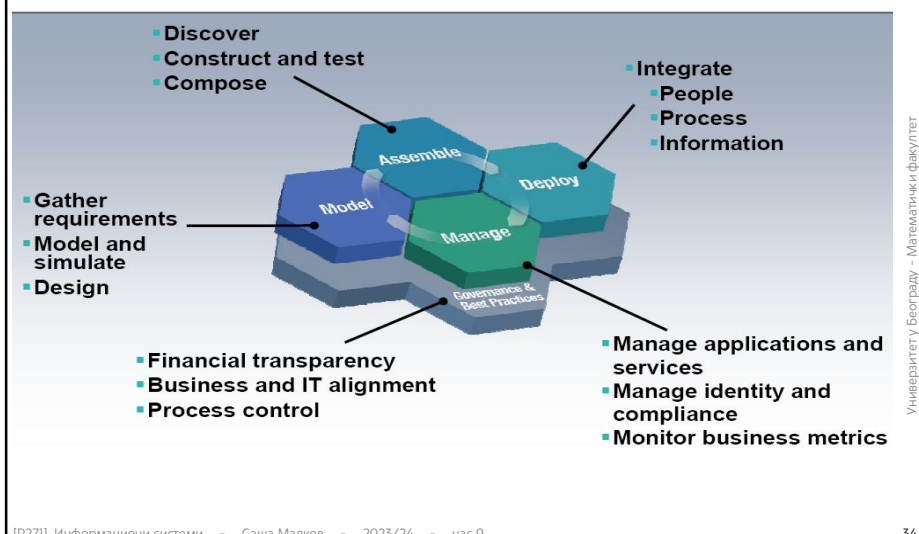
Слојеви архитектуре COA



Референтна архитектура



Животни циклус



Сервисно-оријентисана архитектура / Развој

Улоге у развоју

- Аналитичар пословања (*Business Analyst*)
 - Препознаје и анализира пословне процесе
 - Одговоран за познавање пословних процеса и њихово моделирање
- Интегратор (*Integration Developer*)
 - Интегрише функционалне компоненте организовањем у безбедну, сложу апликацију са архитектуром COA
- Софтверски архитекта (*Software Architect*)
 - Препознаје и пројектује сервисе и архитектуру COA
- Развијач апликација (*Application Developer*)
 - Имплементира компоненте сервиса које су прописане пројектом
 - Препознаје сценарија поновне употребе и развија компоненте које ће се користити при томе
- Администратор система (*System Administrator*)
 - Поставља, конфигурише, усклађује и управља сложеним апликацијама
 - Поставља готове апликације у производно окружење
 - Надгледа извршавање



Имплементација COA

- COA се може имплементирати применом Веб сервиса или других технологија
- код сваке имплементације COA, основна аутономна јединица је *сервис*
- ако то не важи, онда није у питању COA

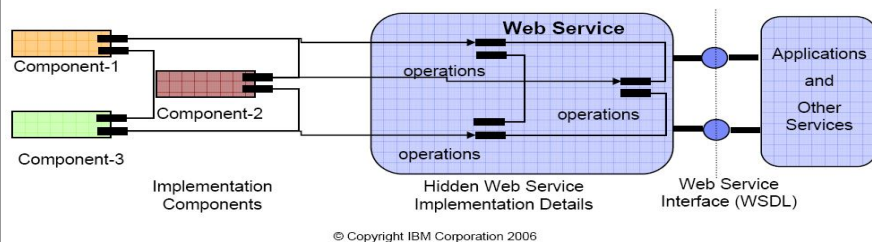


Пример Веб сервиса

- *Web Services Description Language (WSDL)*
 - Језик за описивање Веб сервиса
 - стандард који се користи за описивање Веб сервиса
 - почива на XML-у
 - један од могућих стандарда за описивање сервиса



Пример Веб сервиса (2)



Алтернативе COA

- Постоје бројне алтернативе
- Често се декларишу као COA...
 - ...или као “врста Веб сервиса”...
 - ...али то нису
- Могу бити сасвим добро решење за неке случајеве



Примери алтернатива СОА

- Системи дистрибуираних објеката (*CORBA, COM, ...*)
 - аутономна јединица је *објекат*
- *REST*
 - аутономна јединица је *ресурс*
- Системи за обраду порука
 - аутономна јединица је *ред порука*



Микросервиси

- У последњих неколико година су све популарнији појам *микросервиса* и архитектуре засноване на њима
 - посебно су заступљени у домену ИС мање и умерене величине
 - као и у области веб апликација
 - ублажавају се правила о алатима
 - често се користе скрипт-језици, укључујући варијанте *JavaScript-a*
 - основни циљ је олакшавање остваривања модуларности



Микросервиси и СОА

- Архитектура микросервиса је варијанта *СОА*
- Појам није прецизно дефинисан у пракси
 - често не постоји дубље разумевање односа са *СОА* као ни специфичних предности и недостатака
 - не постоји јасна одредница када *сервис* постаје *микросервис*
 - већина текстова о *СОА* су бољи за теоријско разматрање микросервиса него текстови о микросервисима
- Посветићемо пажњу основним карактеристикама микросервиса
 - сваку од њих могу да имају и класични сервиси
 - али ако их имамо све, онда обично кажемо да је у питању микросервис



Микросервиси

- Неке од основних карактеристика микросервиса су:
 - смањен ниво грануларности
 - одсуство строге формализације протокола
 - децентрализовано управљање
 - протоколи без стања
 - локализовање података
 - аутоматизација развоја и испоручивања
 - толеранција неуспеха



Микросервиси

- смањен ниво грануларности
 - као што сервиси претварају апликације у скуп заокружених функционалних компоненти, тако микросервиси иду корак даље, па се много мање целине обликују као микросервиси
 - оно што би у класичној СОА био један сервис, овде се често дели на више микросервиса
- ...



Микросервиси

- ...
- одсуство строге формализације протокола
 - тражи се лакоћа развоја и пуштања у рад, а посебно реплицирања
 - најчешће се одустаје од сложених протокола и врло често се заснивају на REST-у или чак обичном HTTP-у
 - интерфејси различитих микросервиса у истом окружењу могу да почивају на различитим техникама
- ...



Микросервиси

- ...
- децентрализовано управљање
 - циљ је растерећење трошкова режије
 - остварује се смањивањем међузависности
 - један микросервис обавља заокружени (врло мали) посао, користећи и друге микросервисе, али без претпоставки о остатку система
 - тежи се да се обликовање и имплементација микросервиса раде тако да он што мање зависи од других елемената система али и система као целине
- ...



Микросервиси

- ...
- протоколи без стања
 - често се користе протоколи комуникације без стања
 - одсуство стања олакшава имплементацију и употребу
 - микросервис не мора да чува стање сесије
 - корисник не мора увек да користи исту инстанцу микросервиса
 - олакшава се хоризонтална репликација и повећава скалабилност
 - подстиче се децентрализовано управљање
 - уопштава се дефинисање интерфејса, мада они зато могу да захтевају преношење више података
 - смањује се међузависност микросервиса
- ...



Микросервиси

- ...
- локализовање података
 - као што независно управљање омогућава независан развој, тако независност на нивоу података омогућава преносивост и вишеструку употребљивост
 - остварује се кроз већи број малих колекција података или кроз приступање подацима искључиво преко других микро сервиса који служе само томе
 - то може да представља проблем у домену ИС, зато што се губи целовита обједињена колекција података и отежавају неке врсте послова
 - зато у сложеним ИС велики број микросервиса служи само томе да податке из глобалне колекције (база података) преобликује у формат разумљив неким микросервисима и обрнуто, да податке које неки микросервиси генеришу трансформишу у одговарајући облик и пренесу у базу података
- ...



Микросервиси

- ...
- аутоматизација развоја и испоручивања
 - непрекидна интеграција и непрекидно испоручивање
 - због смањене грануларности, микросервис често не може да буде јединица преносивости (захтева више других микросервиса), али може и уобичајено јесте јединица развоја и испоручивања
- ...



Микросервиси

- ...
- толеранција неуспеха
 - аутоматска замена микросервиса, ако он откаже
 - аутоматска репликација микросервиса, ако је преоптерећен
 - о томе мора да се стара инфраструктура



Микросервиси

- Неке од последица основних карактеристика су:
 - еволуциони животни век
 - микросервиси су јединица развоја и испоручивања али често нису и јединица преносивости
 - низак ниво стандардизације интерфејса може да отежава преносивост
 - не постоји регистар сервиса
 - преносиви су у различита окружења



Микросервиси

- Неке од последица основних карактеристика су:
 - еволуциони животни век
 - микросервиси се одржавају и усавршавају међусобно независно
 - сваки појединачно са одликама еволуционог напредовања
 - због много међузависности, морају да имају стабилан интерфејс
 - када се мења интерфејс, то је често нови микросервис
 - ...



Микросервиси

- ...
- микросервиси су јединица развоја и испоручивања али често нису и јединица преносивости
 - због смањене грануларности, микросервис често не може да буде јединица преносивости (захтева више других микросервиса за успешан рад), али уобичајено јесте јединица развоја и испоручивања
- ...



Микросервиси

- ...
- низак ниво стандардизације интерфејса може да делимично отежа преносивост
 - али једноставност интерфејса (*REST, HTTP, ...*) умањује проблем
 - обично није велики проблем додати нови интерфејс за постојећи микросервис, т.ј. не мора да се имплементира све поново
- ...



Микросервиси

- ...
- не постоји регистар сервиса
 - регистар сервиса има значајну улогу код СОА
 - и код микросервиса се често имплементира нешто као конфигурациони микросервис, који осталима микросервисима пружа неопходне информације о неким (или свим) микросервисима, ако су оне потребе
 - на пример, подаци о микросервису (тј. интерфесу према њему) за приступање подацима, или за плаћање и слично
- ...



Микросервиси

- ...
- преносиви су у различита окружења
 - независност од интерфејса и инфраструктуре чини њихов развој независним од окружења у коме ће се користити
 - практично се изједначава начин развоја за локалне апликације и за масивне апликације у облаку
 - инфраструктура често почива на контејнерима



Будућност микросервиса

- Прилагођенији су агилном развоју од СОА
 - лакше се улази у пројекте, без великог планирања
 - свака целина је независна од осталих целина, али и од система и инфраструктуре
- Флексибилнији су
 - употребљиви су у разним окружењима
 - олакшани су преносивост и скалирање
- Независни су од развојних алата
- Све скупа, чини се да имају светлу будућност



Микросервиси

- James Lewis, Martin Fowler, *Microservices – a definition of this new architectural term*, <http://martinfowler.com/articles/microservices.html>
- Sam Newman, *Building Microservices*, O'Reilly, 2014.