Integracijski načrt

Uvod

Arhitekturna zasnova uporabniškega vmesnika temelji na modulih. Moduli rešitvi doprinesejo funkcionalnosti, ki se lahko razvijajo ločeno in z drugačnimi tehnologijami od ostalih modulov.

Namen

Ta dokument opisuje tehnične zahteve za vgraditev modula v OI PAS.

Dokument je namenjen uporabi v razpisni dokumentaciji za nove in integracijo obstoječih modulov.

Obseg

Dokument opisuje tehnične zahteve in funkcionalnosti za vgraditev modula v OI PAS.

Kratice in okrajšave

Naziv	Opis	Spletni naslov
API	aplikativni vmesnik (Application Programming Interface)	
JWT	JSON Spletni Žeton (JSON Web Token)	https://datatracker.ietf.org/doc/html/rfc7519
KZZ	Kartica zdravstvenega zavarovanja	
OI Ljubljana	Onkološki inštitut Ljubljana	https://www.onko-i.si/

Naziv	Opis	Spletni naslov
OI PAS	Aplikacija za administrativno vodenje in spremljanje pacientov na OI Ljubljana	
On-line	ZZZS ON-LINE	https://www.zzzs.si/zzzs-api/on-line
PK	Profesionalna kartica	
UX	Uporabniška izkušnja (User experience)	
ZZZS	Zavod za zdravstveno zavarovanje Slovenije	https://www.zzzs.si/

Definicije

Klinični modul

Klinični modul (v nadaljevanju: modul) je gradnik sistema OI PAS. Modul v OI PAS sistem prinaša funkcionalnosti, ki s svojo specializacijo podpirajo delovne procese na OI Ljubljana.

Integracija modula v OI PAS se izvaja na nivoju uporabniškega vmesnika. Zaledni sistemi modulov se med seboj ne prepletajo.

Portal

Portal je korenska komponenta OI PAS uporabniškega vmesnika, ki služi kot:

1. Stikalo med samostojnimi moduli.
2. Skupni vir konfiguracije modulov.
3. Most med moduli in sistemskimi funkcionalnostmi (npr. komunikacija z USB napravami).

Relacije med moduli

Moduli se v sistem integrirajo na dva načina, kar določa njihovo vlogo:

1. **Samostojni modul**: Integriran je direktno v Portal in predstavlja ključno, samostojno funkcionalno enoto (npr. modul za naročanje, modul za kartoteko pacienta).

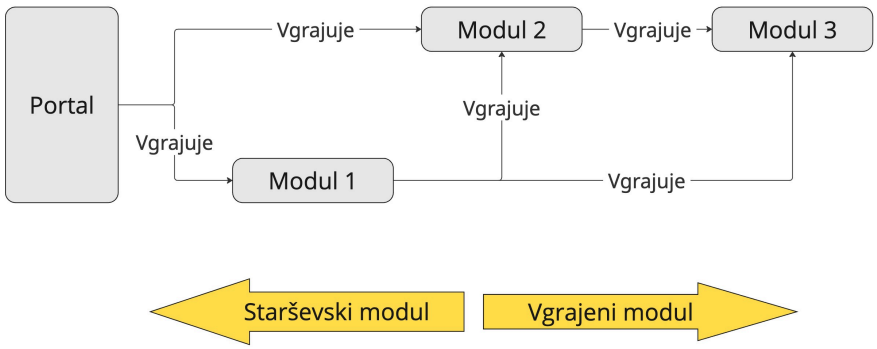
2. **Podporni modul**: Vgrajen je v drug modul (starševski modul) in mu nudi dodatne, podporne funkcionalnosti (npr. modul za izpis nalepk, ki ga lahko kliče več drugih modulov).

Modul, ki integrira drug modul, se v tem kontekstu imenuje **starševski modul**. Modul vgrajen v starševski modul pa se imenuje **vgrajeni modul**.

Če je modul B vgrajeni modul modula A, potem je modul A starševski modul modula B.

Poseben primer modula je Portal, ki je vedno v vlogi starša.

Primer



Zgornja shema prikazuje primer integracije modulov med sabo.

Portal služi kot starševski modul moduloma 1 in 2. Posledično to pomeni da sta modula 1 in 2 Portalova vgrajena modula. Direktna vgraditev v Portal jima dodeli vlogo modula kot samostojnega modula.

Modul 1 vgrajuje modula 2 in 3. Vgraditvi modulov 2 in 3 v modul 1 igrata vlogo podpornega modula.

Modul 3 je zaradi vgraditve tudi podporni modul modulu 2.

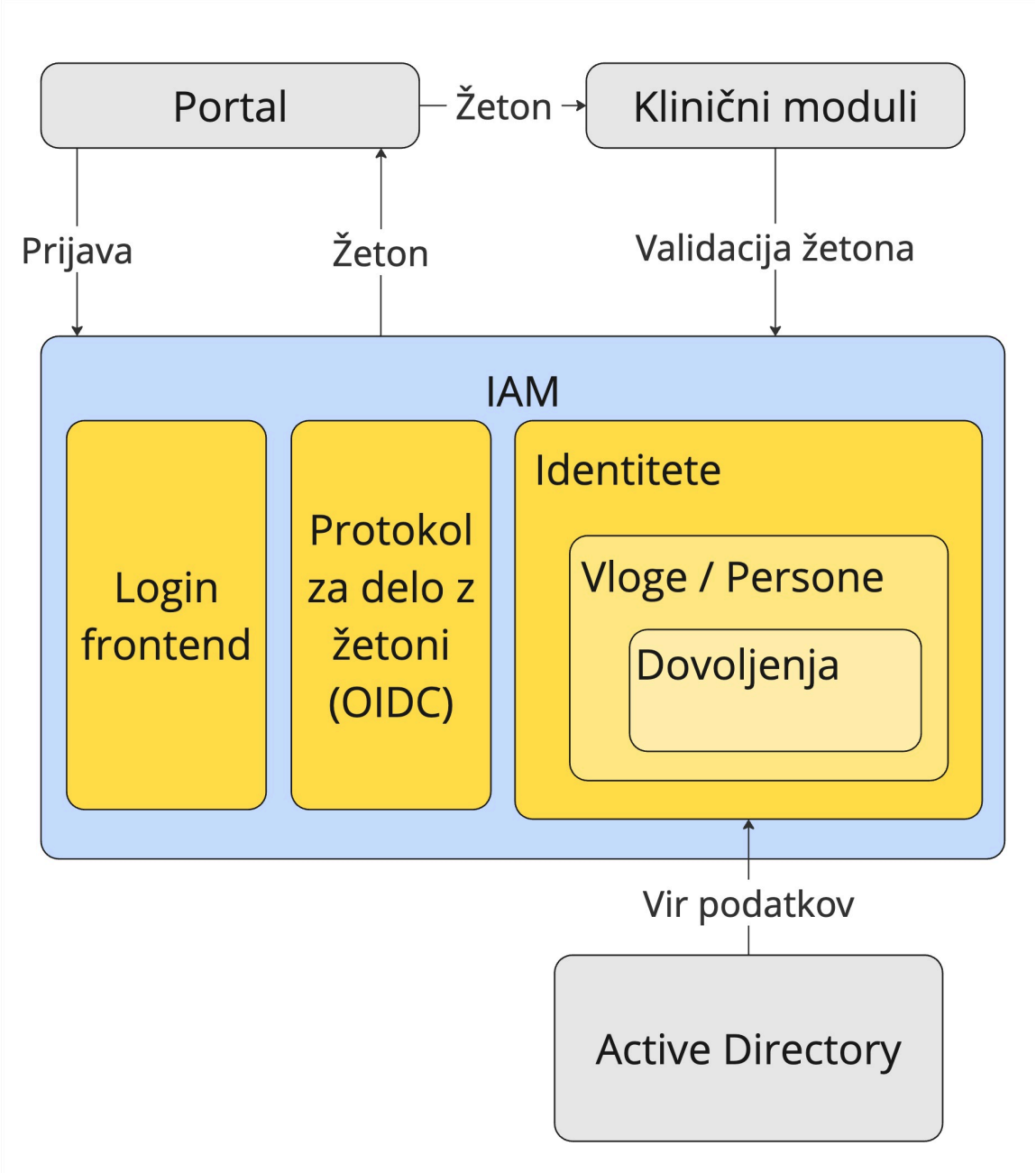
Na podlagi opisa imajo moduli iz primera sledeče vloge:

Modul	Vloge
1	Samostojen modul
2	Samostojen in podporni modul
3	Podporni modul

IAM - Identity and access management

S kratico IAM se sklicujemo na storitev, ki omogoča:

1. Avtentikacijo uporabnikov po `OpenID Connect` shemi.
2. Hranjenje identitete uporabnikov in njihova identifikacija z Active Directory integracijo.
3. Povezovanje uporabnikov z njihovimi personami (npr. zdravnik, strojepiska,...).
4. Povezovanje person z dovoljenji.



Portal

To poglavje podrobneje opisuje funkcionalnosti, ki jih podpira Portal.

Konfiguracija

Modul konfiguracijo sistema (npr. URL posameznega modula) pridobi od Portala med procesom inicializacije. To omogoča nadzor konfiguracije modulov/aplikacije na enem mestu.

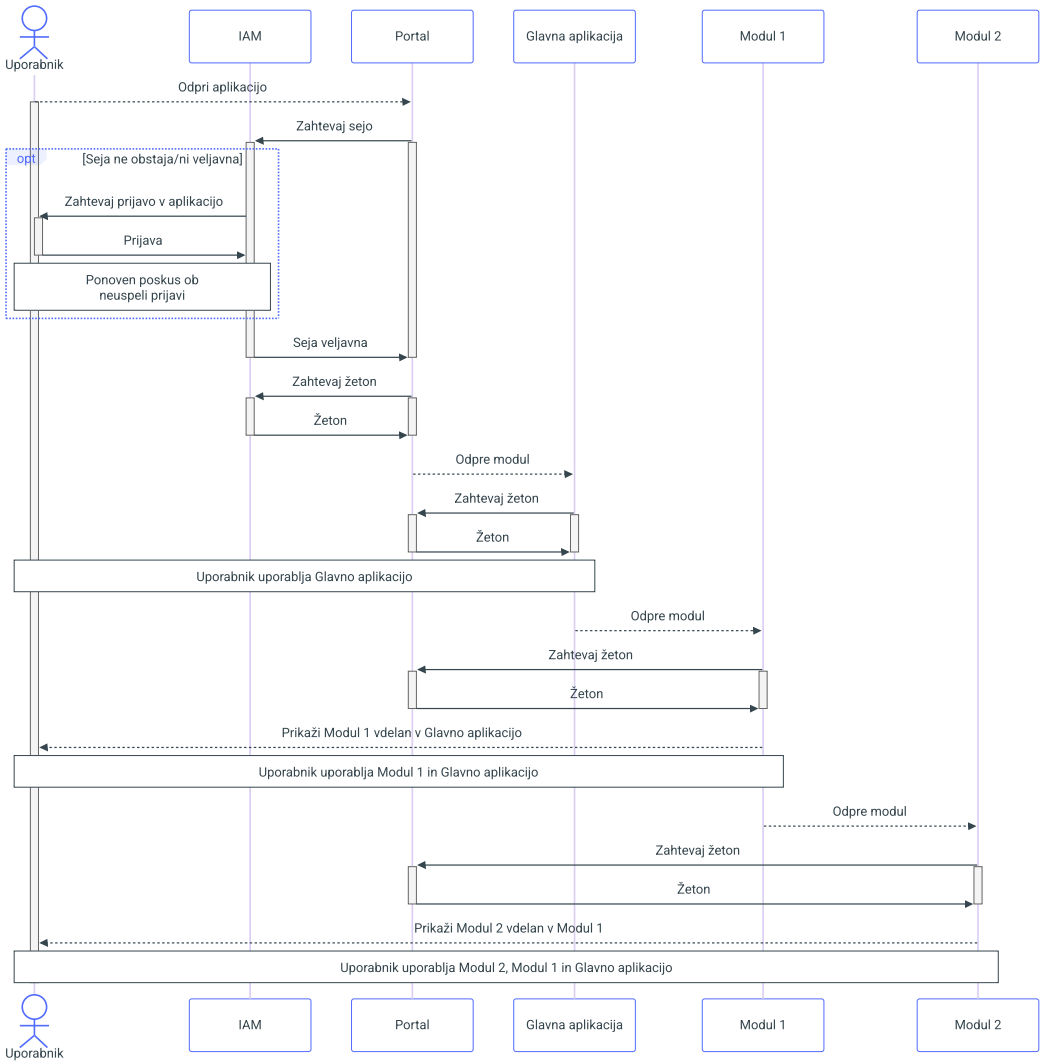
Dostop do native akcij

Portal ima odprt most z native aplikacijo, ki Portalu izpostavi dostop do sistemskih funkcionalnosti. Te so tranzitivno na voljo vsem modulom:

1. ZZZS bralnikov (oz. ZZZS IHIS2 knjižnice).
2. Tiskanja brez potrditvenega okna (silent printing).
3. Čitalcev črtnih in QR kod.
4. Drugih perifernih naprav po predhodnem dogovoru z OI Ljubljana.

Aventikacija

Za enotno izkušnjo tekom uporabe aplikacije se uporabnik vanjo prijavi samo enkrat v Portalu. S prijavo se odpre nova seja na IAM sistemu, ki jo Portal preko žetonov deli z vsemi moduli, ki so vgrajeni vanj.



Koraki:

1. Ob zagonu aplikacije Portal preveri stanje seje z IAM.
    a. V primeru neveljavne/neobstoječe seje uporabnika transparentno preusmeri na prijavni obrazec znotraj IAM-a in izvede prijavo.
2. Portal odpre PAS. PAS ob zagonu od Portala zahteva žeton.
    a. Portal žeton posreduje Glavni aplikaciji in Glavna aplikacija ga uporablja za avtentikacijo s svojim zalednim sistemom.
3. Uporabnik preko Glavne aplikacije odpre Modul 1.
4. Modul 1 od Portala zahteva žeton.
    a. Portal žeton posreduje Modulu 1 in Modul 1 ga uporablja za avtentikacijo s svojim zalednim sistemom.
5. Modul 1 se prikaže in uporabnik ga uporablja skupaj z že prikazano Glavno aplikacijo.
6. Uporabnik prek Modula 1 odpre Modul 2.
7. Modul 2 od Portala zahteva žeton.

    a. Portal žeton posreduje Modulu 2 in Modul 2 ga uporablja za avtentikacijo s svojim zalednim sistemom.
8. Uporabnik uporablja Modul 2 v kombinaciji z Modulom 1 in Glavno aplikacijo.

Naslov portala

Samostojni moduli lahko nastavijo naslov odprtega okna.

Podporna skripta

Za komunikacijo med moduli in odpiranje modulov se uporablja JavaScript knjižnica (ali skripta, odvisno od modulovega toolinga).

Odpiranje modula

Skripta skrbi za odpiranje novih modulov na podlagi konfiguracije pridobljene od Portala in konteksta, ki ga določi starševski modul. Odpiranje modula se dogaja preko `iframe`-a.

Komunikacija med moduli

Tekom odpiranja modula se postavi ogrodje za dvosmerno komunikacijo med vgrajenim in starševskim modulom. Ogrodje omogoča "request-response" pristop k komunikaciji med moduloma.

Request lahko pošlje katerikoli moduli in je namenjen samo enemu od:

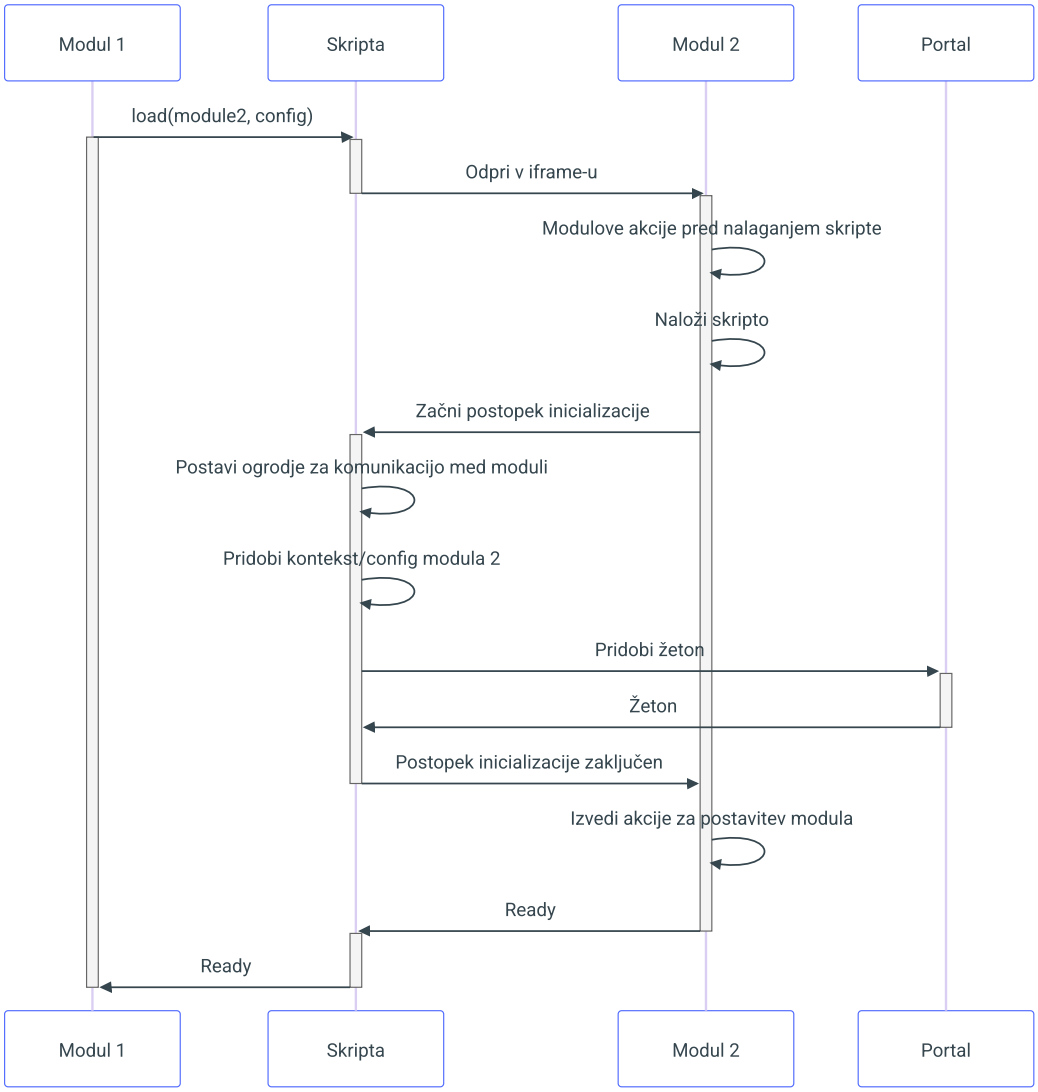
1. Starševski modul
2. Vgrajeni modul
3. Portal

Modul na podlagi potrebne funkcionalnosti, ki jo zahteva starševski ali pa vgrajeni modul definira funkcije, ki jih izpostavi preko skripte. Funkcije, ki so zahtevane iz večih modulov (npr. spremeni pacienta, počisti pogled,...), so abstraktno ponujene s strani skripte. Njihova implementacija pa je prepuščena posameznemu modulu.

Pregled toka odpiranja in komunikacije

Odpiranje

Naslednja shema opisuje postopek odpiranja, nalaganja in inicializacije vgrajenega modula.

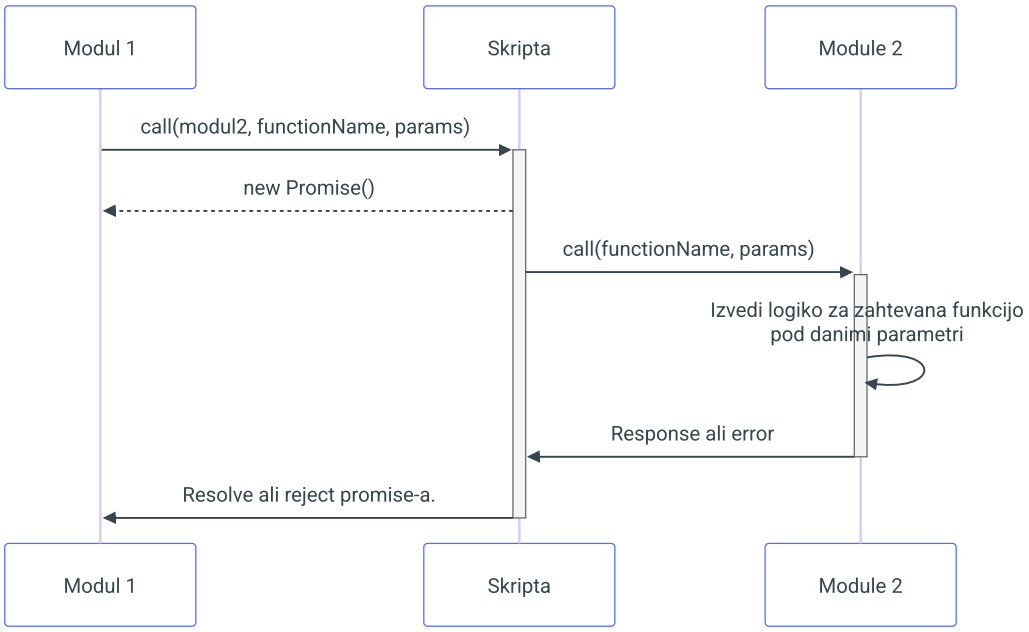


Koraki:

1. Modul 1 pokliče postopek nalaganja Modula 2 in priloži konfiguracijo.
2. Skripta ustvari `iframe` in mu nastavi `src` na Modul 2.
3. Modul 2 (lahko) izvede akcije pred nalaganjem skripte. Konfiguracija in žeton v tem koraku še nista na voljo.
4. Modul 2 naloži skripto in začne postopek inicializacije vgrajenega modula.
    a. Postopek inicializacije postavi ogrodje za komunikacijo med moduli.
    b. Pridobi kontekst/config, ki ga je Modul 1 v točki 1 priložil skripti.
    c. Pridobi avtentikacijski žeton od Portala
5. Modul 2 izvede akcije, ki so potrebne za postavitev modula. Na tej točki lahko uporablja žeton in konfiguracijo.
6. Modul 2 skripti sporoči, da je pripravljen, skripta pa to informacijo poda naprej Modulu 1.

Komunikacija

Spodnja shema opisuje postopek komunikacije med dvema moduloma (bodisi parent>child, bodisi child>parent) ali pa modulom in Portalom (modul>portal)



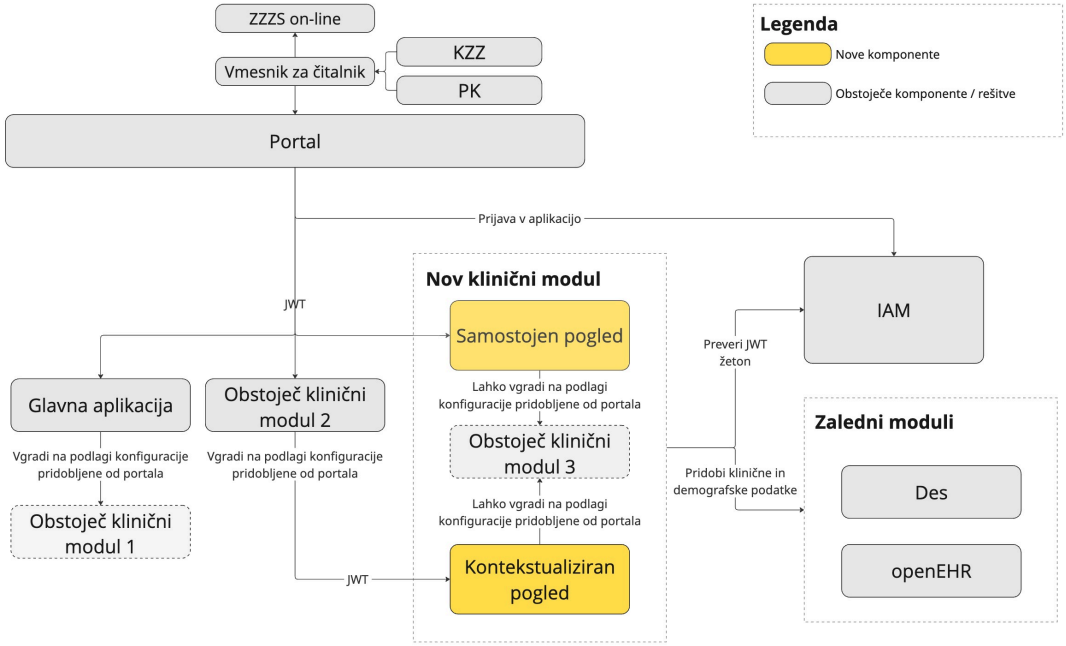
Koraki:

1. Modul 1 skripti sporoči, da želi na Modulu 2 izvesti funkcijo z danimi parametri.
2. Skripta Modulu 1 vrne Promise objekt, Modulu 2 pa posreduje zahtevek Modula 1.
3. Modul 2 izvede zahtevano funkcijo.
4. Modul 2 skripti vrne rezultat funkcije. Rezultat je lahko response ali pa error.
5. Skripta rezultat posreduje nazaj Modulu 1 preko Promise objekta.

Integracija modula

> **Namen**
>
> Ta razdelek opisuje zahteve, ki so obvezne za vgraditev modula.

Naslednja shema orisuje lokacijo novega modula v OI PAS rešitvi na OI Ljubljana.



Z rumeno barvo so označene komponente, ki so del razvoja novega modula. S sivo barvo so označene komponente, ki že obstajajo.

Vir in shranjevanje podatkov

Modul je primoran uporabljati že obstoječe rešitve na OI Ljubljana za shranjevanje demografskih in kliničnih podatkov.

Demografski podatki se pridobivajo iz in shranjujejo v Better FHIR R4 strežnik.

Klinični podatki se pridobivajo iz in shranjujejo v Better OpenEHR strežnik.

Uporaba drugih virov podatkov (npr. baza za uporabniške nastavitve) ali specializirane baze (npr. time-series baze) je dovoljena ob predhodni potrditvi s strani OI Ljubljana.

Določitev pogleda

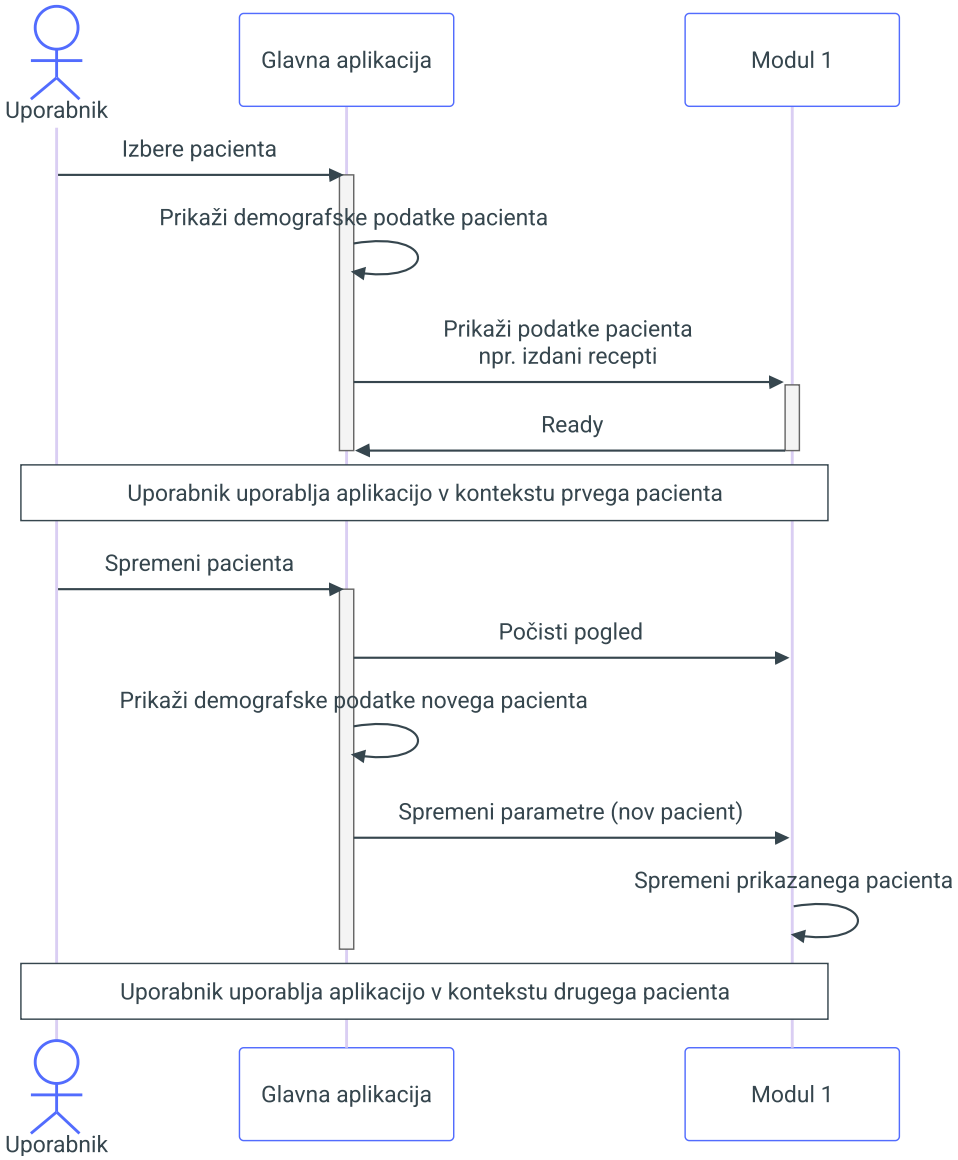
Odvisno od namena modula lahko ta omogoča različne poglede. Na primer, modul za recepte lahko ponuja različna pogleda za:

1. Pregled vseh pacientu izdanih zdravil.
2. Obrazec za izdajo novega zdravila.

Starševski modul zahteva različne poglede glede na kontekst vgraditve modula. Pogled se določi:

1. Preko konfiguracije ob vgraditvi modula.
2. Naknadno z uporabo podporne skripte.

Tekom uporabe lahko starševski modul spremeni parametre pogleda z uporabo podporne skripte.



Parametri, ki jih vgrajeni modul sprejme se določijo v dogovoru z izvajalcem starševskega modula na podlagi njihovih potreb.

Sporočanje dogodkov

Vgrajeni moduli sporočajo dogodke starševskim modulom. Dogodke, ki jih vgrajeni modul sporoča, se določi v dogovoru z izvajalcem starševskega modula na podlagi njegovih potreb.

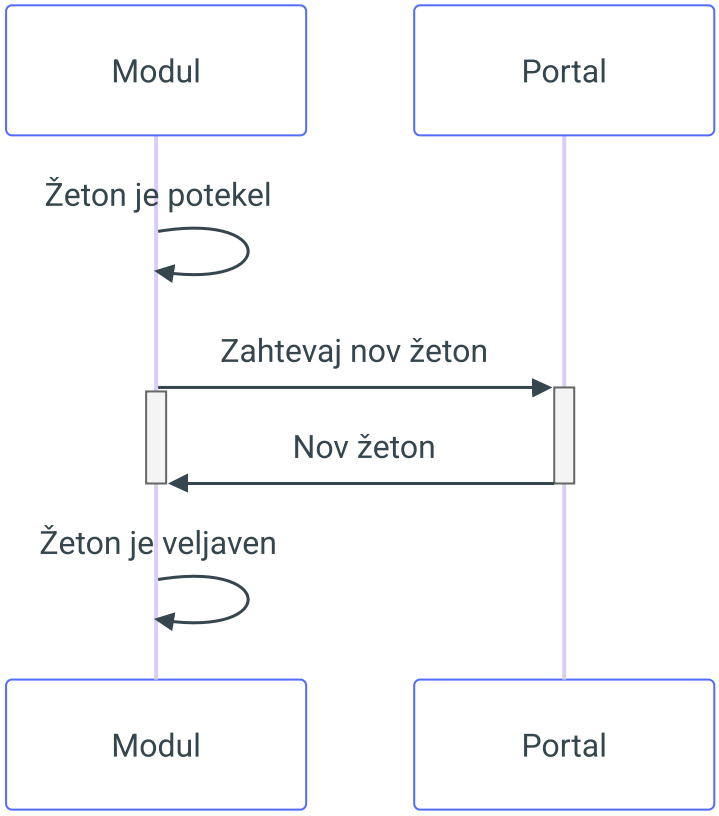
Primeri dogodkov:

1. Ustvarjeno je bilo laboratorijsko naročilo.
2. Recept je bil izdan.
3. Uporabnik je shranil obrazec.
4. ...

Zahtevanje novega žetona

Modul je zadolžen, da spremlja veljavnost žetona oz. implementira mehanizem, ki zazna, da žeton ni več veljaven. Če žeton poteče od Portala zahteva nov veljaven žeton.

Modul mora poskrbeti, da se zaradi morebitnega uporabljanja pretečenega žetona podatki ne izgubijo. Logika pridobivanja novega žetona mora biti za uporabnika transparentna/nevidna.



Preverjanje žetona

Modul mora ob vsakem zahtevku skladno z `OpenID Connect` protokolom z IAM sistemom preveriti veljavnost žetona, ki ga modul uporablja. Za preverjanje uporabi JWKS mehanizem, ključe si shrani v short-lived cache in tako minimalno obremenjuje IAM strežnik.

Avtorizacija uporabnika

Žeton vsebuje uporabnikove avtorizacijske podatke (vloge in pravice).

Seznam vlog in pravic

Modul pripravi seznam vseh vlog in pravic, ki se jih prednastavi znotraj IAM-a. Skrbnik modula je dolžan spremembe (dodajanje, odstranjevanje, ...) sporočiti skrbniku IAM-a in se uskladiti glede načina posodobitve modula v vseh okoljih.

Večjezikovna podpora

Modul mora podpirati izbiro jezika v procesu nalaganja modula. Jezik določi starševski modul v posredovani konfiguraciji, ki je prejeta ob nalaganju modula.

Varnostne zahteve

1. **Preverjanje izvora sporočil**: Pri komunikaciji med moduli preko `postMessage` API-ja mora vsak modul obvezno preveriti izvor (`event.origin`) sporočila in sprejeti le sporočila iz Portalove domene.
2. **Validacija vnosov**: Modul je odgovoren za validacijo in čiščenje (sanitization) vseh uporabniških vnosov.

Obravnava napak in beleženje

1. **Beleženje**: Modul mora implementirati mehanizem za beleženje napak (logging). Vsi kritični dogodki in napake morajo biti zabeleženi.
2. **Prikaz napak uporabniku**: Napake, ki se prikažejo uporabniku, morajo biti razumljive in ne smejo vsebovati občutljivih tehničnih podrobnosti.

Verzioniranje

1. **Verzija API-ja modula**: Vsak modul mora jasno opredeliti verzijo svojega API-ja, ki ga izpostavlja preko `GET /version` HTTP endpointa.
2. **Upravljanje nezdružljivih sprememb**: Morebitne nezdružljive spremembe (breaking changes) v API-ju modula morajo biti usklajene z vsemi odvisnimi moduli in OI Ljubljana.

Revizijska sled (audit log)

Aplikacija mora sama skrbeti za revizijsko sled.

V primeru uporabe cache-a mora modul zagotoviti, da se v revizijsko sled zapišejo dostopi do pacientovih podatkov, ki so v cache-u.

Dobavitelj modula mora na zahtevo OI Ljubljana zagotoviti dostop do zahtevanih zapisov iz revizijske sledi.

Health check

Modul mora imeti HTTP endpoint z uporabo katerega lahko IT sistem na OI Ljubljana periodično preverja status delovanja modula.

Obračun

Za posredovanje podatkov za obračun je odgovoren modul. Modul sporoča dogodke, ki so relevantni za obračun (npr. "opravljena storitev X").

Pridobivanje podatkov drugega modula

Kadar modul potrebuje podatke od drugega modula, s katerim ni v direktni starš-otrok relaciji, komunikacija poteka preko Portala, ki deluje kot posrednik (message broker). Modul A pošlje zahtevek Portalu (npr. `portal.call('moduleB', 'getData', params)`), ki ga ta posreduje Modulu B. To preprečuje direktno vezavo med moduli in ohranja arhitekturno čistost.

Priporočila pri razvoju modula

Iskanje pacienta

Pogledi, ki kot parameter sprejmejo pacientov ID, naj ne omogočajo iskanja pacienta.

V izogib spremembi pacienta SAMO znotraj modula, ki je vgrajen v pacientov pogled naj vgrajeni modul ne omogoča menjave pacienta v kolikor tega ne zahteva starševski modul. Primer takega modula je seznam pacientovih izdanih receptov ali pacientovih čakajočih naročil.

Opomba: Pogledi, ki niso namenjeni vgradnji na nivo pacienta lahko omogočajo iskanje pacienta.

Uporabniška dokumentacija

Izvajalec modula lahko priskrbi uporabniško dokumentacijo v Markdown formatu. Ta dokumentacija bo vključena v skupek OI PAS dokumentacije in dana na voljo uporabnikom aplikacije na OI Ljubljana.

UX napotki

Napotki za oblikovanje UX-a aplikacije se nahajajo v prilogi Integracijski načrt: UX.