

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Aplikačné architektúry softvérových systémov

RETAILOVÝ E-SHOP TELEKOMUNIKAČNÉHO OPERÁTORA

Semestrálne zadanie

Autori: Bc. Michal Farkaš

Bc. Matej Kováč

Bc. Jozef Zatlko

Vypracované: letný semester 2015/2016

1 Úvod

Cieľom tejto práce je návrh architektúry informačného systému pre retailový e-shop telekomunikačného operátora. Práca sa venuje architektonickým pohľadom Rozanského a Woodsa. Modely sú vytvorené prevažne v jazyku UML. Práca obsahuje opisy týchto modelov a použitých architektonických princípov.

2 Požiadavky na systém

Telekomunikačný operátor má svoj e-shop, cez ktorý predáva svoje mobilné služby, internetové služby a káblovú televíziu. Tieto zahŕňajú aj predaj zariadení k týmto službám: mobilných telefónov rôznych typov, notebookov, routerov, set-top boxov a iných zariadení.

Požiadavkou na retailový e-shop je, aby poskytoval webové rozhranie a používateľské obrazovky boli vysoko interaktívne. Z uvedeného je využitý v návrhu architektúry JavaScript MVC framework AngularJS.

2.1 Funkcionálne požiadavky

- štandardná funkcionálna „nákupného košíka“
- objednanie tovaru vrátane potvrdenia
- platba za tovar vrátane potvrdenia
- predaj paušálov vrátane súčastí:
 - volania v rámci siete toho istého operátora
 - volania do iných sietí
 - SMS/MMS
 - internet v mobile
 - volanie blízkym osobám
 - paušál na zahraničné volania (do vybraných krajín)
 - poistenie telefónu
- predaj internetových balíkov
 - podľa objemu dát
 - podľa rýchlosti sťahovania/odosielania
- predaj rôznych typov balíkov televíznych programov
- poskytovanie zľavnených cien podľa balíkov služieb

2.2 Nefunkcionálne požiadavky

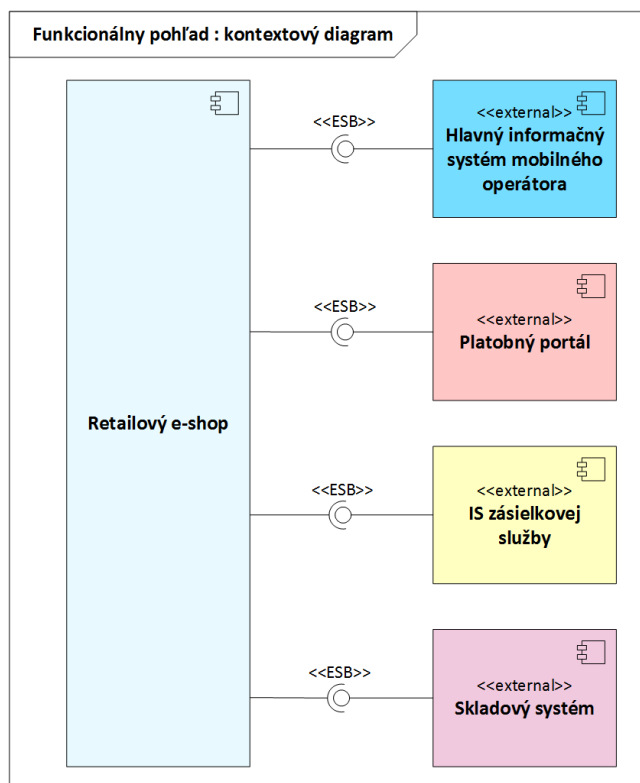
- Spoľahlivosť
- Bezpečnosť
 - identifikácia a autentifikácia klienta
 - bezpečný platobný portál
- Dostupnosť
 - zálohovanie
 - replikácia dát
 - vyrovňovanie záťaže
- Škálovateľnosť a výkonnosť
- Modifikovateľnosť
- Prepojitelnosť s externými systémami
- Použitelnosť webového rozhrania

3 Funkcionálny pohľad

Opisuje najdôležitejšie elementy systému z pohľadu poskytovanej funkcionality - ich zodpovednosť, rozhrania a vzájomnú interakciu.

3.1 1. úroveň: Kontextový diagram

Opisuje systém ako celok a jeho spoluprácu s externými systémami.



Obrázok 1: Funkcionálny pohľad: kontextový diagram

3.1.1 Opis externých komponentov

Hlavný informačný systém mobilného operátora - poskytuje informácie o klientoch, ich zriadených službách

Platobný portál - zabezpečuje elektronický spôsob zaplattenia za zakúpené produkty, okrem post-paid služieb (paušálov)

IS zásielkovej služby - poskytuje najmä informácie o aktuálnom stave zásielky v prípade, že si zákazník objedná niektoré zo zariadení, ktoré mu je doručované zásielkovou službou

Skladový systém - slúži pre evidenciu tovaru, najmä množstva a dostupnosti jednotlivých zariadení

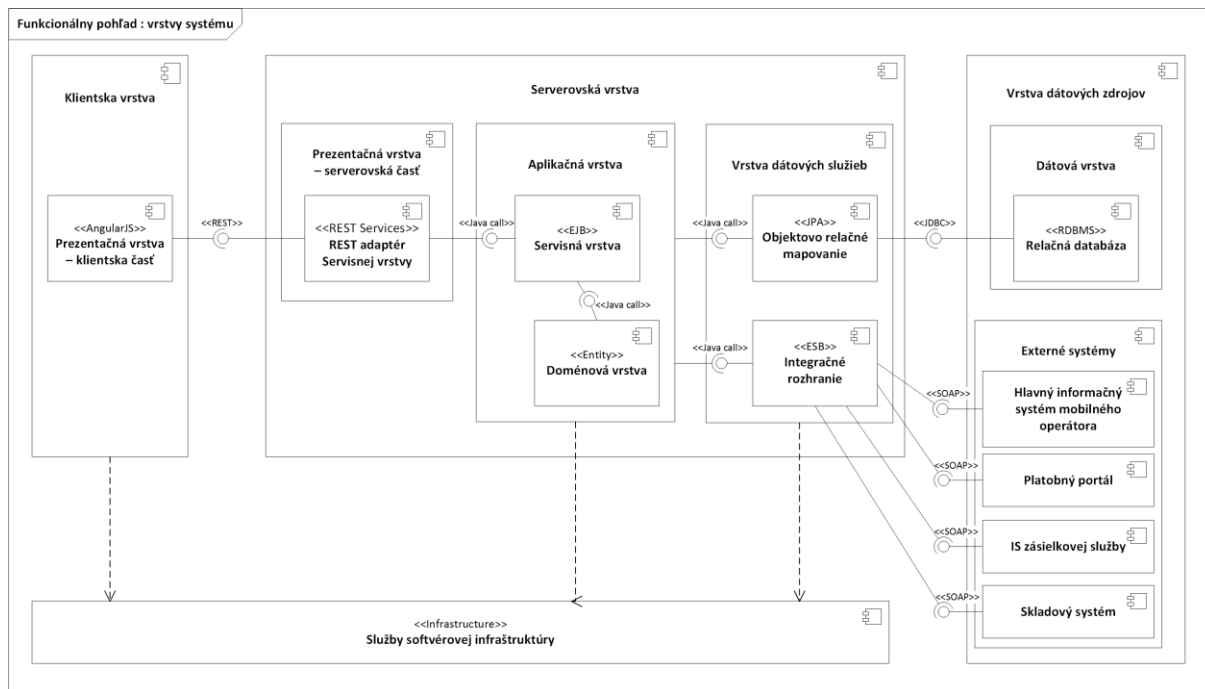
3.1.2 Použité architektonické princípy

ESB (Enterprise Service Bus) - je model softvérovej architektúry využívaný pri navrhovaní a realizácii komunikácie medzi vzájomne sa ovplyvňujúcimi softvérovými aplikáciami v architektúre orientovanej na služby (SOA). Predstavuje zbernicu rôznych služieb, t.j. softvérový architektonický model pre distribuovanú komunikáciu medzi aplikáciami. Jeho primárnym použitím je jednotná integrácia viacerých podnikových aplikácií s odlišným prostredím.

3.2 2. úroveň: Vrstvy systému

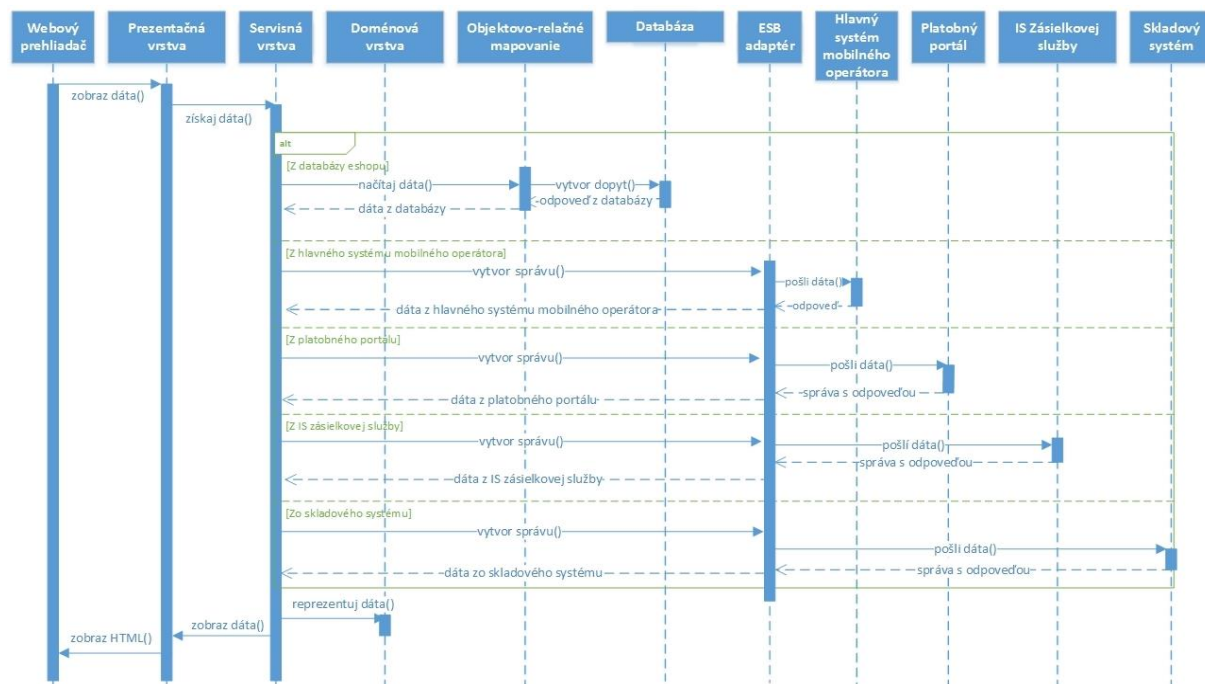
Opisuje systém z pohľadu architektonického štýlu vrstvy.

3.2.1 Funkcionálny pohľad: vrstvy systému



Obrázok 2: Funkcionálny pohľad vrstvy systému

3.2.2 Funkcionálny pohľad: komunikácia vrstiev



Obrázok 3: Funkcionálny pohľad: komunikácia vrstiev

3.2.3 Zodpovednosti jednotlivých vrstiev

Prezentačná vrstva - Zodpovedá za zobrazovanie dát používateľovi, poskytuje komponenty používateľského rozhrania. Slúži ako vstupný bod pre komunikáciu používateľa so systémom. Delí sa na klientsku časť (JavaScript, AngularJS) vykonávanú vo webovom prehliadači a na serverovskú časť (Servlet).

Aplikačná vrstva - Zabezpečuje spracovávanie požiadaviek používateľa. Obsahuje aplikačnú logiku.

Vrstva dátových služieb - Zabezpečuje prepojenie s vrstvou dátových zdrojov.

Vrstva dátových zdrojov - Zabezpečuje prístup k relačnej databáze. Zodpovedá za vkladanie, výber, aktualizácie a odstraňovanie dát z databázy.

3.2.4 Použité architektonické princípy

Architektonický štýl vrstvy - Tento architektonický štýl rozdeľuje systém na vrstvy, ktoré sú špecifikované na základe úrovni abstrakcie. Vrstva predstavuje zoskupenie elementov, kde elementy s príbuznou funkcionalitou sa nachádzajú v spoločnej vrstve. Používanie vrstiev podporuje oddelovanie záležitostí a umožňuje tak lepšiu pružnosť a udržiavateľnosť systému.

JavaScript MVC - Model-Pohľad-Ovládač (angl. Model-View-Controller) je architektonický štýl, ktorý sa používa pri architektúre prezentačnej vrstvy systému. Pozostáva z troch častí: model, ktorý je zodpovedný za údržbu dát, pohľad zodpovedný za zobrazovanie dát používateľovi a ovládač, ktorý má na starosti komunikáciu medzi modelom a pohľadom.

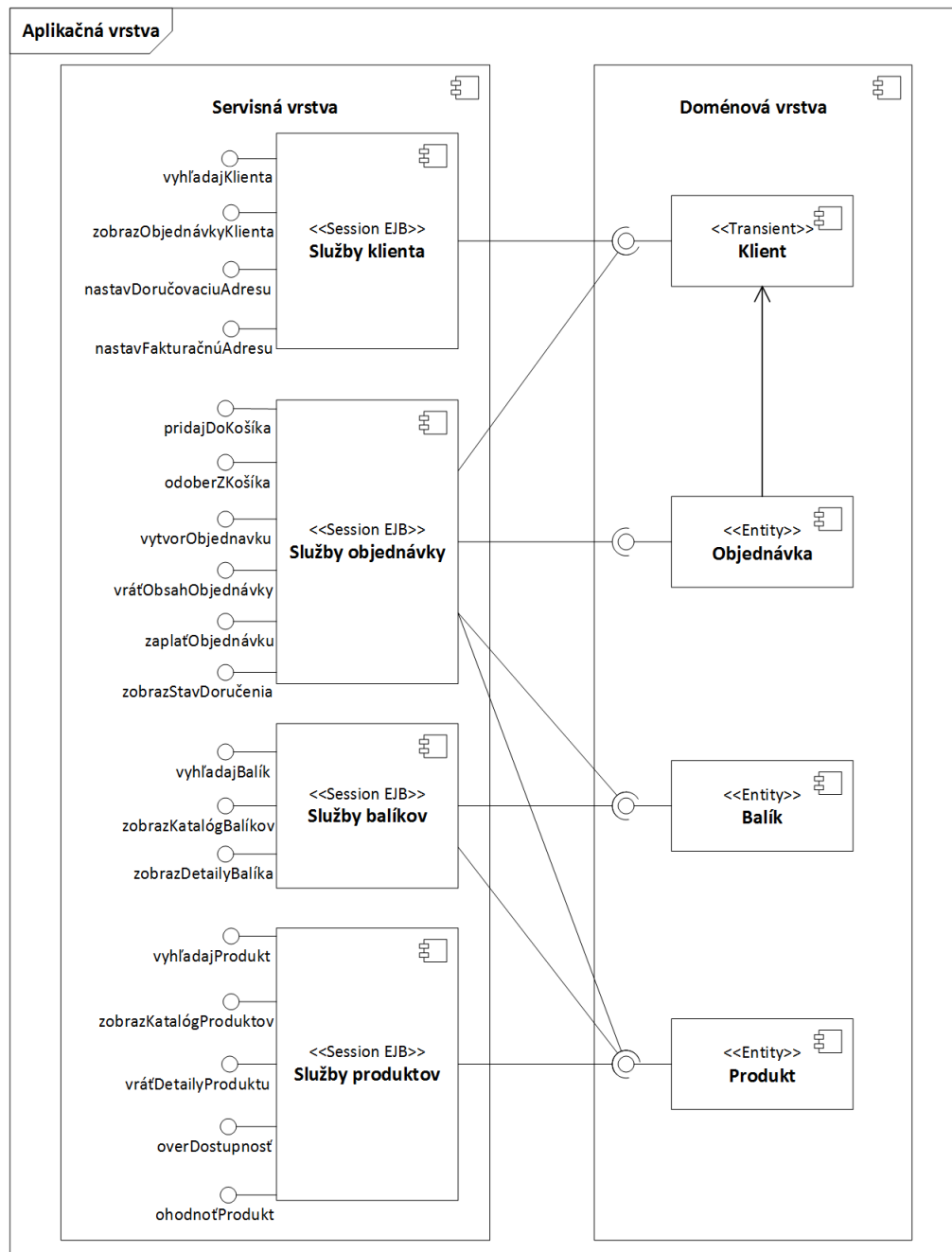
REST (representational state transfer) - REST je architektonický štýl založený na rozhraniach poskytujúcich služby, ktoré je možné zavolať jednoduchou HTTP požiadavkou.

SOAP (Simple Object Access Protocol) - SOAP je protokol slúžiaci pre výmenu štruktúrovaných informácií s využitím implementácie webových služieb. Pre výmenu dát používa formát XML a protokoly aplikačnej vrstvy ako HTTP, prípadne SMTP.

JPA (Java Persistence API) - Je špecifikácia rozhraní pre správu dát v relačných databázach pre platformu Java. Predstavuje iba špecifikáciu a tým pádom pre použitie vyžaduje použitie implementácie, napríklad Hibernate, EclipseLink, TopLink a iné.

3.3 3. úroveň: Komponenty aplikačnej vrstvy

Zobrazuje členenie aplikačnej vrstvy na servisnú a doménovú vrstvu a komponenty v nich obsiahnuté.



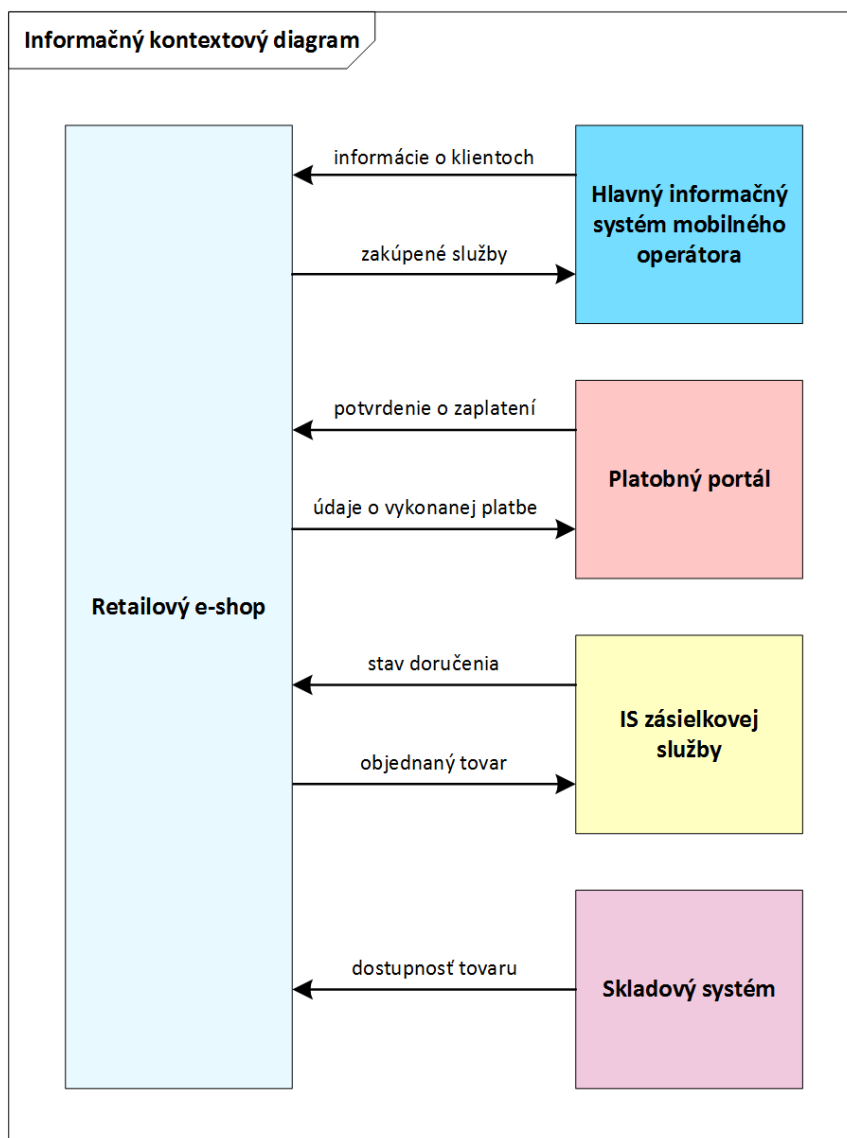
Obrázok 4: Funkcionálny pohľad: komponenty aplikačnej vrstvy

4 Informačný pohľad

Špecifikuje štruktúru informácií systému, spôsob ich ukladania, manipulácie, správy a distribúcie.

4.1 Kontextový diagram toku dát

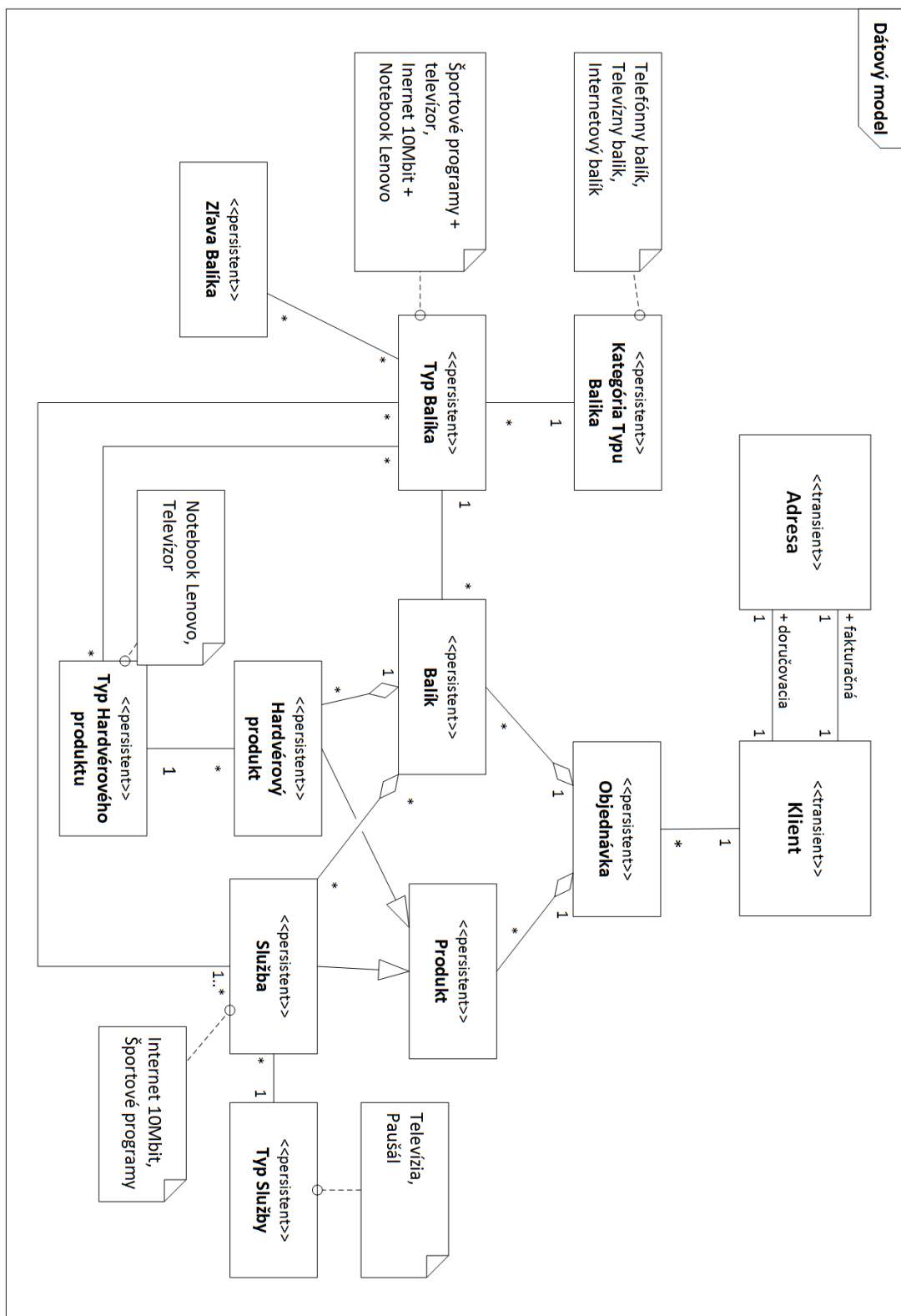
Ukazuje typy dát, ktoré prúdia medzi systémom a jeho externými systémami



Obrázok 5: Informačný pohľad: kontextový diagram toku dát

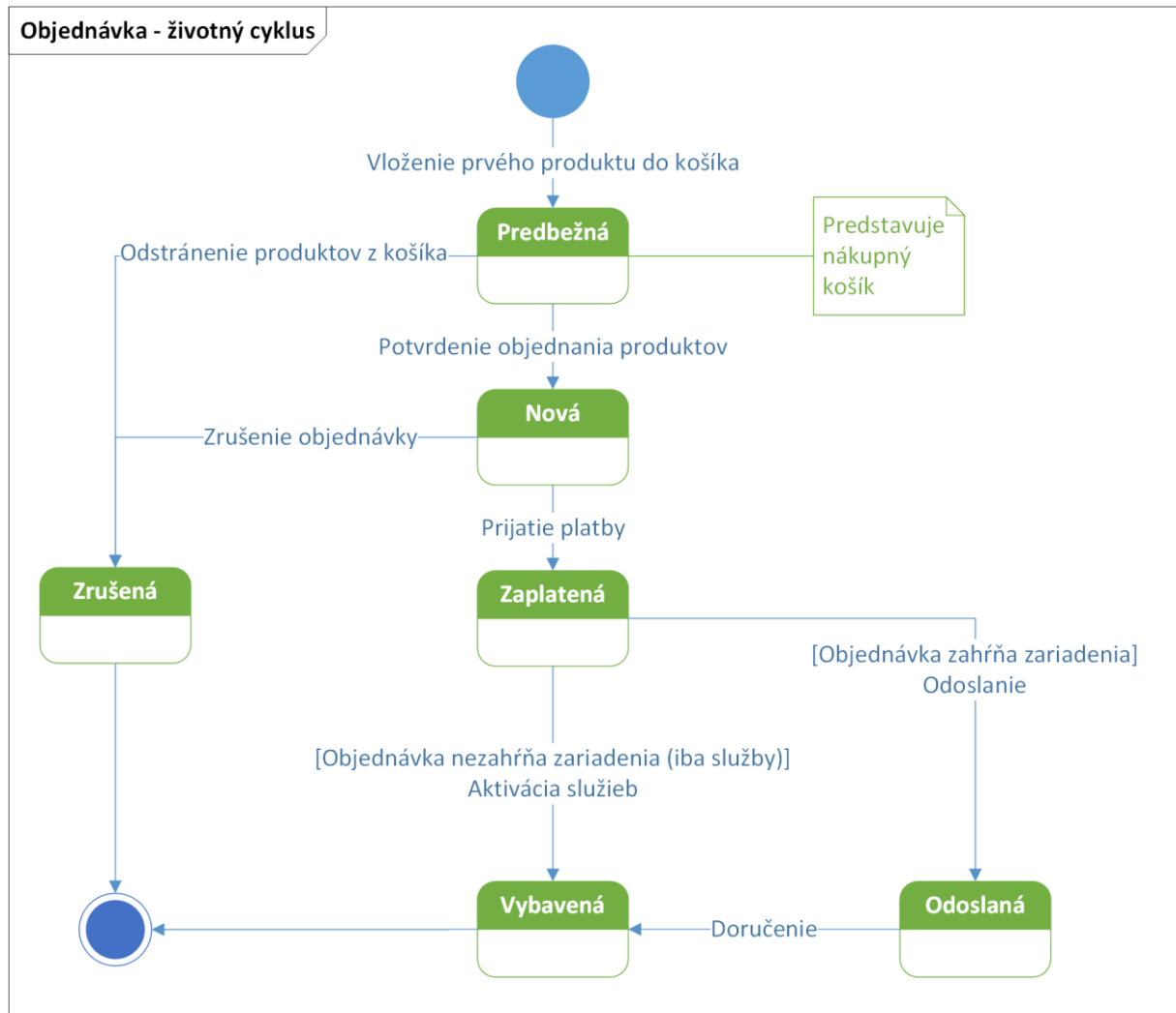
4.2 Dátový model

Modeluje štruktúru dát v e-shope mobilného operátora.



Obrázok 6: Informačný pohľad: dátový model

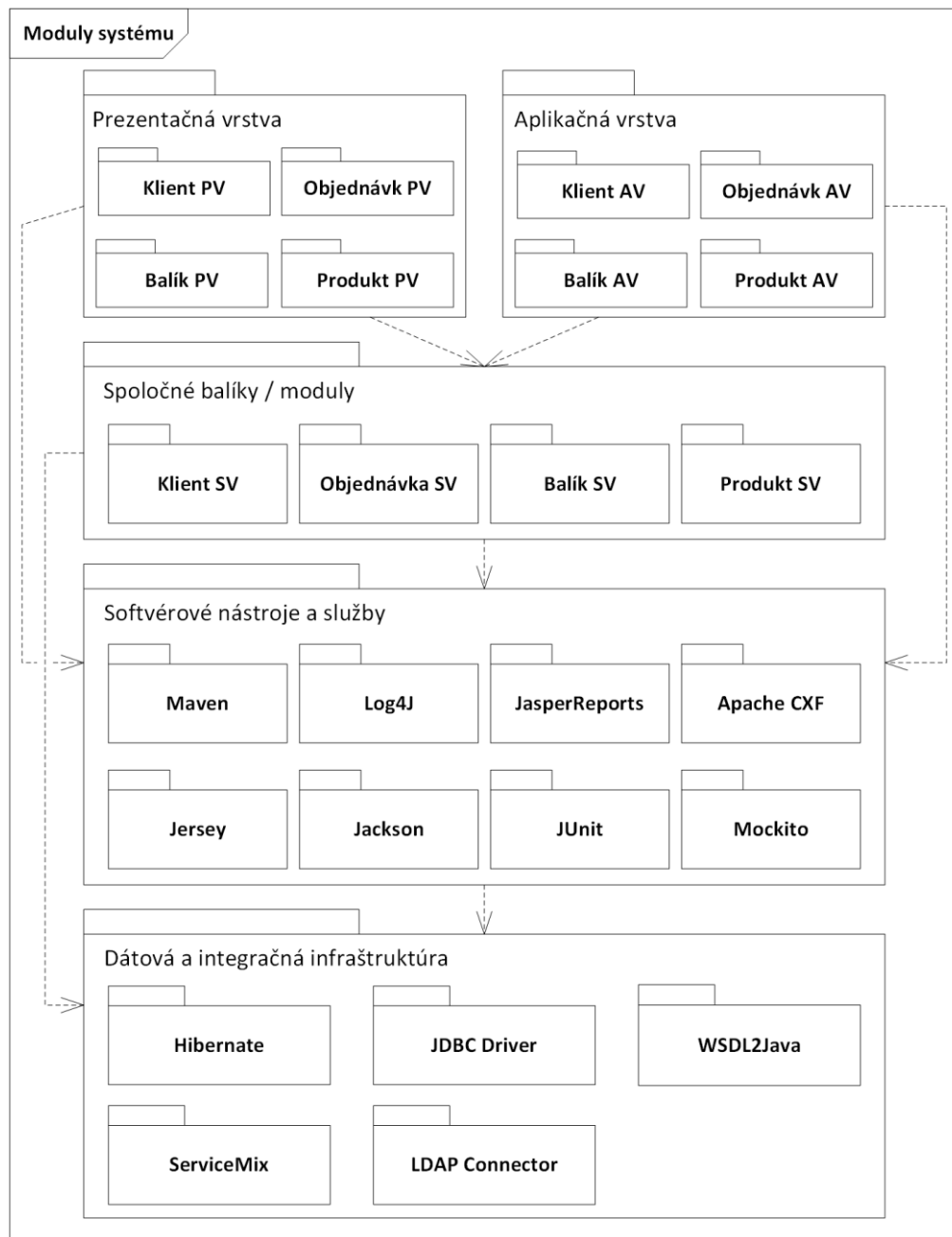
4.3 Model životného cyklu: Objednávka



Obrázok 7: Model životného cyklu: Objednávka

5 Vývojový pohľad

Zobrazuje základné moduly systému - programový kód, ich štruktúru, vzájomné vzťahy a závislosti. Zahŕňa potrebné softvérové nástroje a softvérové súčasti základnej infraštruktúry.

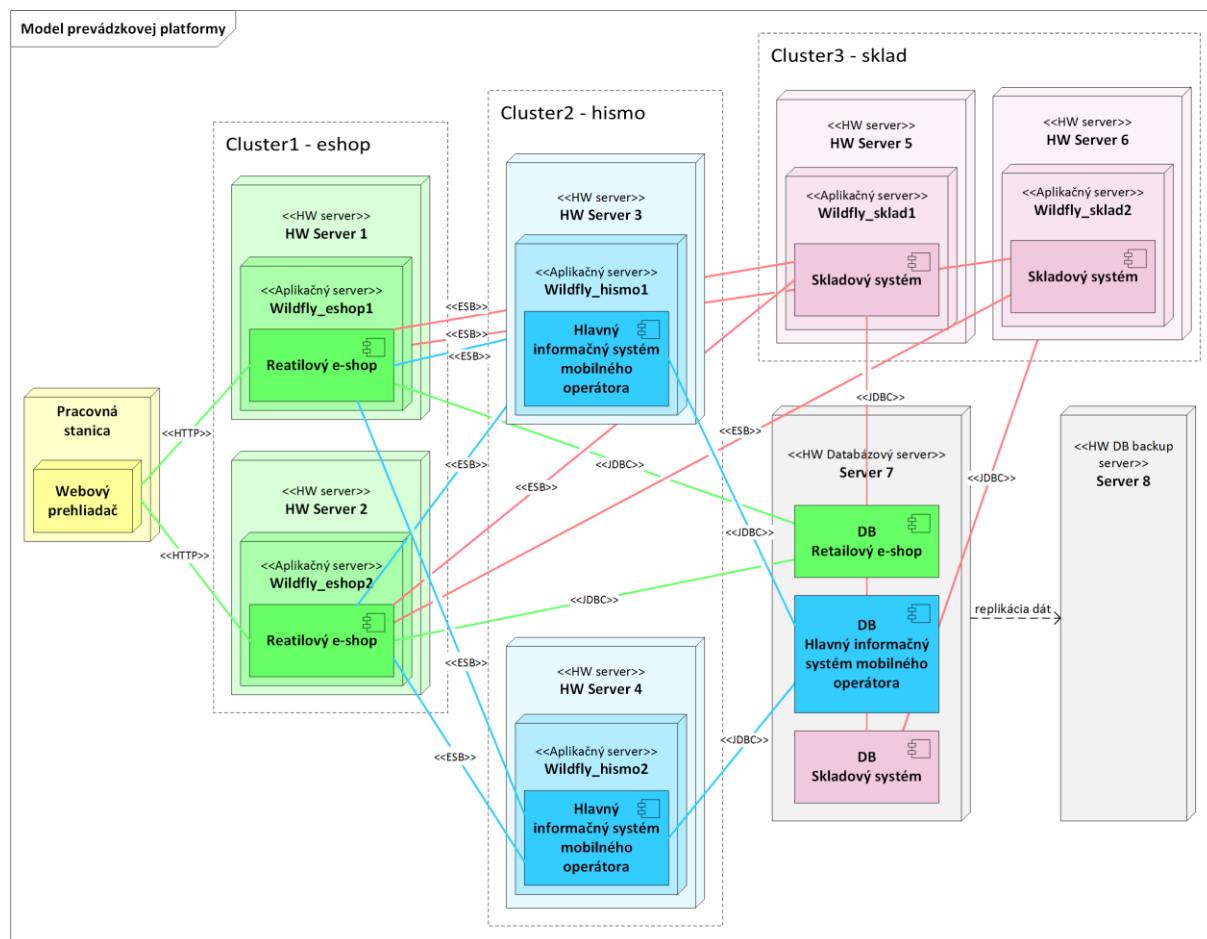


Obrázok 8: Vývojový pohľad: moduly systému

6 Pohľad rozmiestnenia elementov

Špecifikuje na aké hardvérové, softvérové a sieťové elementy bude systém v prevádzke umiestnený.

6.1 Model prevádzkovej platformy



Obrázok 9: Pohľad rozmiestnenia elementov: model prevádzkovej platformy

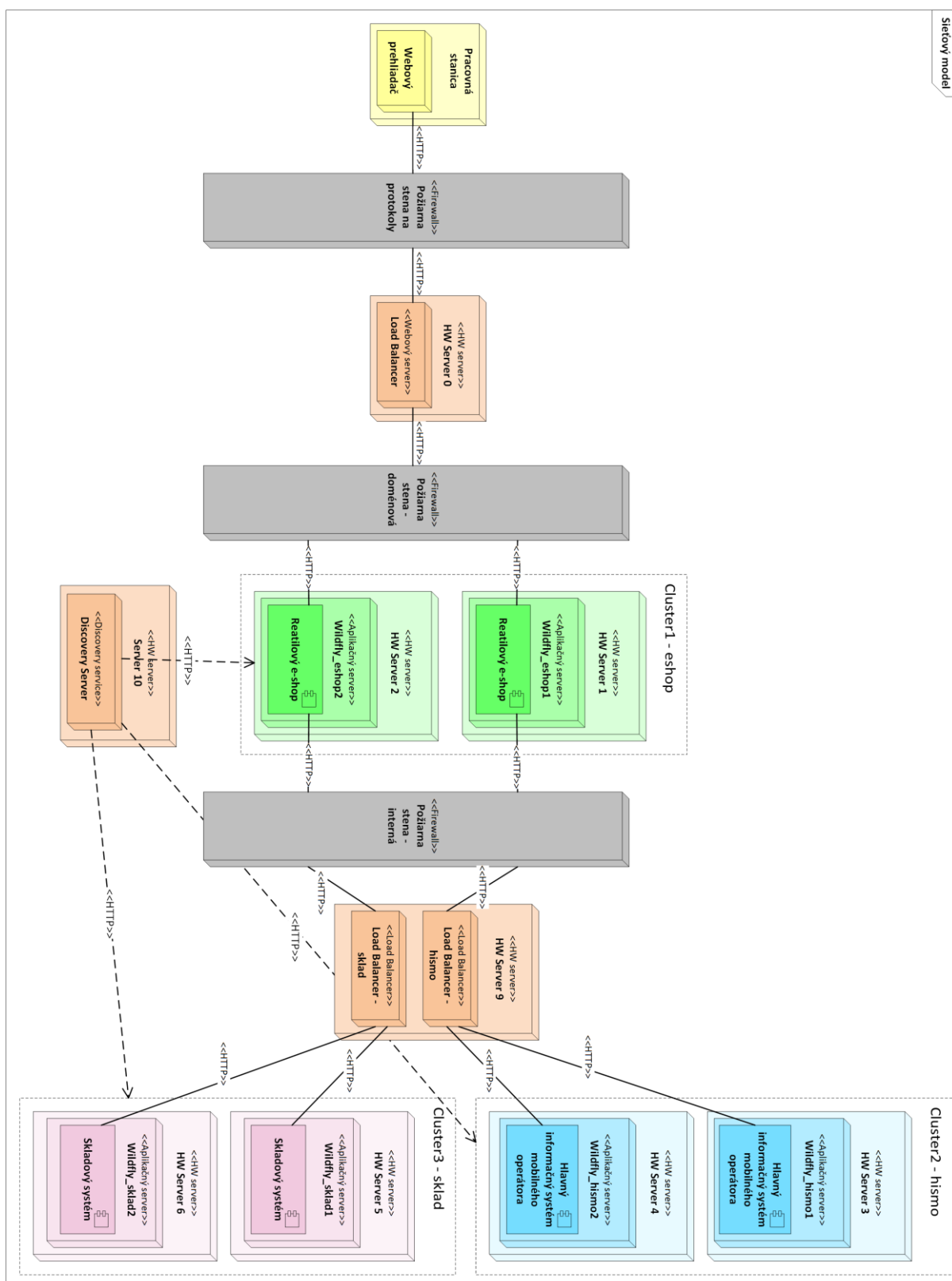
6.1.1 Použité architektonické princípy

Distribúovanie systémov - architektonický princíp, pri ktorom sú komponenty rozmiestnené na viacerých strojoch, ktoré sú vzájomne prepojené sieťou a komunikujú výmenou správ. Jednotlivé komponenty vzájomne interagujú za účelom dosiahnutia spoločného cieľa. Účelom distribúcie je horizontálne škálovanie systému a vyrovnanie sa s výpadkom jedného systému.

Vyrovňavanie záťaže (Load Balancing) - technika pre rozloženie záťaže medzi dva alebo viacero strojov (prípadne iných zariadení) s cieľom optimálneho využitia priepustnosti so zlepšením doby odozvy. Použitie viacerých strojov pri vyrovňovaní záťaže môže tiež poskytovať lepšiu odolnosť proti výpadkom.

Zálohovanie (replikácia) databázových dát - spočíva vo vytvorení kópie dát z databázy. Vytvorená kópia sa uloží na iný stroj ako pôvodné dáta, vďaka čomu sa môže použiť pre (čiastočnú) obnovu stratených dát v prípade neočakávanej udalosti ako výpadok, odcudzenie a podobne.

6.2 Sieťový model



Obrázok 10: pohľad rozmiestnenia elementov: sieťový model

6.2.1 Použité architektonické princípy

Firewall (požiarna stena) - Predstavuje sieťové zariadenie alebo softvér, ktorého úlohou je oddeliť siete s rôznymi prístupovými právami (typicky napr. Internet a Intranet) a kontrolovať tok dát medzi týmito sieťami.

Load Balancer (vyrovnávač záťaže) - Zariadenie slúžiace pre rozloženie záťaže medzi dva alebo viacero strojov (prípadne iných zariadení) s cieľom optimálneho využitia priepustnosti so zlepšením doby odozvy. Použitie viacerých strojov poskytuje lepšiu proti výpadkom.

Service Discovery – Ide o službu, ktorá zisťuje dostupnosť a správne fungovanie služieb v systéme. Pri výpadku, alebo inom probléme upozorní na tento fakt vyrovnávače záťaže. Tie potom prestane smerovať požiadavky na túto nefunkčnú službu. Toto je umožnené vďaka použitiu distribuovaných systémov.

7 Záver a zhodnotenie

Na základe špecifikácie zadania sme navrhli architektúru systému. Tá predstavuje abstraktný návrh, ktorý je ešte potrebné špecifikovať. Pri návrhu sme použili architektonické pohľady Rozanského a Woodsa. Modely sme vytvorili prevažne v jazyku UML s využitím nástroja Microsoft Visio 2013.

V predstavených modeloch sme sa snažili zachytiť požiadavky zo zadania a uplatniť súčasné architektonické princípy.