

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Návrh a technologická analýza prostřednictvím Sales Cloud
Diplomová práce

Autor: František, Konečný
Studijní obor: Aplikovaná informatika

Vedoucí práce: doc. Ing. Hana Tomášková, Ph.D.

Hradec Králové

Květen 2023

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 3.5.2023

František Konečný

Poděkování:

Děkuji vedoucí diplomové práce doc. Ing. Hana Tomášková, Ph.D. za odborné vedení, konzultace a čas strávený nad mojí prací.

Anotace

Diplomová práce se zabývá problematikou obměny CRM v prostředí reálně existující firmy. Jejím ústředním cílem je popsat analýzu procesu obměny a řešení napříč jednotlivými fázemi implementace. Diplomová práce je rozdělena do dvou částí, a to teoretické a praktické. V teoretické části autor popisuje CRM systém Salesforce – Sales Cloud, migrace a integrace, které úzce souvisí s danou problematikou.

Praktická část práce je zaměřena na formalizaci problematiky a popis jednotlivých řešení, které v rámci implementace projektu nastaly.

V závěru práce shrne autor výsledky, které implementace nového CRM uživatelům přinesla.

Annotation

Title: Design and technological analysis through Sales Cloud

Diploma theses is focused on problematics of an existing company which needed replacing CRM system. The main theme is to describe analysis which was used in the process of replacement through each phase of the implementation. Thesis is divided into two parts – theoretical and practical. Author is describing CRM system – Sales Cloud and its migration and integration capabilities in the theoretical part.

Practical part is focused on formalizing the problematics of implementing such a software and description of individual solutions, which were needed thought the project.

The author will discuss the solution which was implemented in the last part of the thesis.

Obsah

1	Úvod.....	1
2	Cíl práce.....	3
3	Teoretická část	4
3.1	Co je CRM?	4
3.1.1	Obecně o Salesforce	4
3.2	SalesForce	8
3.2.1	Standardní objekty.....	9
3.2.2	Record Type.....	11
3.2.3	Object pipeline – Životní cyklus objektu.....	12
3.2.4	Validační pravidla.....	12
3.2.5	Uživatelské Role	13
3.2.6	Uživatelské Profily	13
3.2.7	Permission Set – Sada povolení	14
3.2.8	Speciální pole	14
3.2.9	Back End.....	18
3.2.10	FrontEnd.....	25
3.3	Problematika migrací	26
3.3.1	ETL a Migrační nástroje	27
3.3.2	Salesforce a jeho migrační nástroje.....	27
3.4	Integrace v rámci Salesforce	28
3.4.1	Architektura Integrací v Salesforce	28
4	Praktická část.....	29
4.1	O Společnosti.....	30
4.2	Business procesy	31
4.2.1	Vznik Zákazníka	32

4.2.2	Vznik provozovny.....	33
4.3	První fáze.....	33
4.3.1	Představení starých systémů.....	34
4.3.2	Business procesy – první fáze.....	37
4.3.3	Integrace systémů – první fáze.....	43
4.4	Druhá fáze.....	51
4.4.1	Rozdělení starého a nového CRM.....	52
4.4.2	Problematika nového produktového katalogu v Salesforce a nemožnost jeho synchronizace 1:1 se Siebel.....	54
4.4.3	Rekonciliace entit aneb zpětná cesta ze Siebel do Salesforce.....	55
4.4.4	Představení nových systémů.....	56
4.4.5	Business procesy – druhá fáze.....	56
4.4.6	Integrace systémů – druhá fáze.....	67
4.5	Třetí fáze.....	73
4.5.1	Legacy Adapter.....	74
4.5.2	Business procesy – třetí fáze.....	74
4.5.3	Integrace systémů – třetí fáze.....	79
5	Shrnutí výsledků.....	88
6	Závěry a doporučení.....	90
6.1	Návrh na zlepšení dosavadního CRM.....	90
7	Seznam použité literatury.....	91

Seznam obrázků

Obrázek 1 Chování Apex	19
Obrázek 2 Integrovní architektura.....	29
Obrázek 3 Zákazník z ARES (první fáze) – vytvořeno autorem.....	38
Obrázek 4 Vytvoření Kontraktu ze Siebel (první fáze) – vytvořeno autorem.....	40
Obrázek 5 Vytvoření Kontraktu z OMS (první fáze) – vytvořeno autorem.....	40
Obrázek 6 Vytvoření INP (první fáze) – vytvořeno autorem.....	41
Obrázek 7 Business model (první fáze) – vytvořeno autorem.....	42
Obrázek 8 Vytvoření Workshopu (první fáze) – vytvořeno autorem	43
Obrázek 9 Conversation diagram provázanosti systémů v první fázi – vytvořeno autorem	44
Obrázek 10 Integrovní toky entit při operaci ADD (první fáze) – vytvořeno autorem	45
Obrázek 11 Integrovní toky entit při operaci EDIT (první fáze) – vytvořeno autorem	46
Obrázek 12 Zákazník z ARES (druhá fáze) – vytvořeno autorem.....	58
Obrázek 13 Zákazník, adresa a kontaktní osob avytvořena manuálně (druhá fáze) – vytvořeno autorem.....	59
Obrázek 14 Vytvoření Workshopu, adresy a vazby na INP + rekonciliace do Salesforce (druhá fáze) – vytvořeno autorem	60
Obrázek 15 Flow pro zasmluvnění (druhá fáze)	65
Obrázek 16 Vznik smlouvy (druhá fáze) – vytvořeno autorem.....	67
Obrázek 17 Conversation diagram provázanosti systémů ve druhé fázi – vytvořeno autorem	68
Obrázek 18 Integrovní toky entit při operaci ADD (druhá fáze) – vytvořeno autorem	69
Obrázek 19 Integrovní toky entit při operaci EDIT (druhá fáze) – vytvořeno autorem	69
Obrázek 20 Vytvoření Workshopu, adresy a vazby na Contract Line (třetí fáze) – vytvořeno autorem.....	76
Obrázek 21 Vytvoření kontraktu (třetí fáze) – vytvořeno autorem	77

Obrázek 22 Vytvoření Rekonfigurační nabídky (třetí fáze) – vytvořeno autorem ..	79
Obrázek 23 Conversation diagram provázanosti systémů ve třetí fázi – vytvořeno autorem	80
Obrázek 24 Integrované toky entit při operaci ADD (třetí fáze) – vytvořeno autorem	81
Obrázek 25 Integrované toky entit při operaci EDIT (třetí fáze) – vytvořeno autorem	82

1 Úvod

Každá společnost, která využívá informační technologie se jednou ocitne v situaci, kdy musí zhodnotit svou technologickou stránku a zjistit, zda dané systémy dostačují pro její efektivní fungování.

Efektivita IT systémů je v dnešním „digitálním věku“ velice významná a vede k potencionální výhodě na trhu. Do efektivit IT systémů se řadí jak výkonost daných platforem, tak způsob komunikace mezi systémy. Nedílnou součástí jsou pak dobře nastavené business procesy, které, pokud jsou správně zkombinované s IT strukturou, mohou kýženého zlepšení efektivit dosáhnout.

Pokud se bavíme o společnosti, která uplatňuje klasické modely fungování, tedy B2B (Business to business) či B2C (Business to Customer), je pravděpodobné, že bude potřebovat CRM systém, a to alespoň v nějaké formě.

Pokud je společnost mladá, není třeba implementace robustních CRM, které mají možnost rozšíření se do korporátních rozměrů. Většinou se tam zvolí CRM, které je momentálně dostačující. Pokud se však společnost rozroste a dané CRM či jiné software nejsou už vyhovující, řeší se právě jejich obměna.

Obměna CRM není však výsadou rostoucích firem. Mnohdy je třeba tento proces výměny podstoupit čistě z praktických důvodů, jako je bezpečnost, chybějící funkcionality nebo nedostatečná škálovatelnost stávajícího software.

Pokud však k výměně dojde, je to často složitý proces, který si vyžaduje mnoho příprav. Tato diplomová práce si klade za cíl ukázat jednu z podobných implementací a navrhnout a popsat postup řešení.

Záměrem této diplomové práce je snaha o seznámení čtenáře se systémem CRM Salesforce – Sales Cloud. Budou popsány jednotlivé prvky jeho struktury. Dále se bude autor snažit namapovat toto CRM na stávající řešení. Celá implementace bude rozdělena do několika fází, přičemž v každé fázi budou podrobně analyzovány provedené změny a jejich dopad bude komparován s výchozí situací.

Diplomová práce je podložena nejen teoretickým pozadím fungování CRM Salesforce, ale také empirickým výzkumem a osobní praktickou zkušeností autora. Příklad zvolené firmy vychází z praxe reálného procesu implementace. Práce

poskytne náhled do skutečného fungování firemních software se všemi jejich komplikacemi.

2 Cíl práce

Cílem této práce je na vybraném projektu efektivně formalizovat problémy, které mohou nastat při nasazení nového CRM do společnosti. Tyto vznikající komplikace budou formalizovány pomocí BPMN 2.0 diagramů. Dále se autor zaměří na vzniknuvší problémy a pokusí se popsat jejich řešení. V rámci práce se budou mapovat procesy uvnitř systémů celé společnosti a budou skrze více fází projektu postupně popsány. Ústředním tématem práce je návrh a řešení efektivnější implementace Salesforce Sales Cloud pro konkrétní společnost.

Práce rovněž představí aplikaci Legacy Adapter, který slouží jako nástroj pro načtení, transformaci a přesun dat a která byla jako jedno z řešení implementována. Dalším cílem práce je vytvoření case study, kterou bude možno využít při podobných implementacích projektů, jako byl tento.

3 Teoretická část

3.1 Co je CRM?

CRM – Customer relationship management. Obecně se jedná se o systém, který spravuje všechny vztahy mezi společnostmi a jejich zákazníky či potenciálními zákazníky[1]. CRM je systém umožňující efektivně třídit a vyhodnocovat informace o zákaznících a obchodních aktivitách společnosti, která CRM využívá. CRM systémy napomáhají shromažďovat detailní informace o aktuálních zákaznících a jejich potřebách, využívaných službách či produktech, které kupují. Dále rovněž eviduje veškeré informace o obchodních nebo i finančních transakcích – fakturace, objednávky, poptávky, nabídky atd. Některé CRM podporují zákaznické služby např. helpdesk (servisní centrum) nebo se umí napojit a komunikovat s konkrétními podnikovými službami, zejména již existující systémy, které vykonávají v rámci společnosti různé činnosti. Neopominutelným přínosem CRM systémů je jejich vliv na usnadnění analýzy zákazníka, plánování dalších obchodních kroků nebo tvorbě strategie v oslovování pomocí marketingových kanálů. Pokud je CRM správně využíváno má pozitivní dopad na analýzu efektivního rozvoje obchodních i marketingových služeb. CRM systém shromažďuje veškerá data, která o zákaznících máme. Ty nám pomáhají zákazníkům a jejich potřebám lépe porozumět a jednotlivé obchodní procesy v naší firmě těmto potřebám přizpůsobit. A to ať jde o model B2B, tedy obchodní vztah a vzájemnou komunikaci mezi obchodními společnostmi[2 s. 11] nebo B2C , kdy jde o vztah mezi obchodní společností a koncovým zákazníkem[3 s. 11].

CRM je informačním systémem spravující údaje týkající se interakce se zákazníkem a bývá propojen poté s dalšími IS jako je marketing, účetnictví, fakturační systém atd. [3].

3.1.1 Obecně o Salesforce

Platforma Salesforce byla založena roku 1999 Marcem Benioffem[4] s dalšími třemi zakladateli. Jejich vize byla „*A World-Class Internet Company for Sales Force Automation*“ což lze přeložit jako Světovou internetovou společnost pro

automatizaci obchodních procesů. Už od začátku chtěli, aby jejich systém fungoval jako tehdejší internetová stránka Amazon.com [5 s. xvii]. Rozhodli se tedy pro technologii, kterou nyní nazýváme Cloud¹[6], což mělo veliký vliv na další vývoj systému. Přístup společnosti Salesforce k problému CRM, že žádná firma není totožná a nelze mít tedy stejný systém pro všechny, se stal revolučním. Salesforce podporuje snadné přizpůsobení systému zákazníkovi za pomoci vytváření nových objektů a jejich atributů, vazeb, podrobného nastavení rolí a oprávnění, standartními Application Programming Interface – API²[7] nebo obchodem s rozšířeními zvaného Appexchange³[8]. Salesforce zpoplatňuje své služby formou zprostředkování licencí pro každého uživatele. Jedná se o předplatné, které se platí každý měsíc, za každou aktivní licenci, kterou má společnost k dispozici [9]. Na komunitu Salesforce administrátorů, programátorů a specialistů v různých odvětvích, ve kterých má Salesforce zastoupení formou integrovaných aplikací a dalších řešení, je kladen velký důraz.

3.1.1.1 Sales Cloud

Sales Cloud je produkt společnosti Salesforce, který se zaměřuje na obchod a veškeré procesy s ním spjaté. Salesforce dokáže velice jednoduše automatizovat procesy ve firmě a tím zrychlovat odbavení zákazníka. Sales Cloud, jakožto konkrétní aplikace Salesforce, je určená pro obchodníky. Jejím hlavním cílem je nabídnout důležité informace o zákazníkovi zaměstnancům firmy. Tyto informace se musí nacházet na jednom místě, a to co nejpřehledněji, tak aby obchodník nemusel nic složitě dohledávat. Mezi základními informacemi je například detail zákazníka, tzv. Client 360 neboli Klientská třistašedesátka, kde se na jedné stránce nachází potřebné informace, jako např. otevřené obchodní příležitosti, kontaktní

¹ Termínem cloud v IT rozumíme jistý popis globální sítě na sobě nezávislých a funkcionálně diferencovaných serverů. Jinak řečeno - jedná se o rozsáhlou síť vzájemně propojených vzdálených serverů fungujících jako koherentní systém.

² API neboli pro Application Programming Interface je v oblasti softwarového inženýrství chápán jako rozhraní pro programování aplikací. Jedná se o soubor postupů, funkcí, tříd či programových celků, které má programátor k dispozici a k užívání. API rovněž určuje způsob, jímž jsou funkce knihovny volány ze zdrojového kódu programu.

³ Jedná se o obchod s doplňky do Salesforce, kde lze najít více než 5000 řešení, které vám pomůžou rozšířit Salesforce v jakémkoliv oboru nebo průmyslu.

osoby nebo historie komunikace, aktivit a událostí. Jeden zákazník nemusí být vždy v kompetenci jednoho obchodníka. Sales Cloud nabízí Account Team (Tým zákazníka), kde se dají nastavit práva k přístupu informacím pro jednotlivé členy. Sales Cloud nebude příliš účinný nástroj bez individualizace aplikace konkrétní firmě. Jinými slovy nelze si zakoupit licence Sales Cloud a začít aplikaci v plné míře okamžitě používat. Měla by proběhnout úprava Sales Cloud tak, aby zákazníkovi přinesla výhody. Aby Salesforce byl co nejdostupnější a přinesl svým uživatelům výhodu na trhu, je k dispozici prostřednictvím mobilní aplikace, která dokáže fungovat i off-line nebo např. pomocí rozšíření do Microsoft Outlook⁴[10]. Všechna tato prostředí se musí však přizpůsobit, a to nejen firmě jako celku, ale i konkrétním skupinám uživatelů a jejich rolím.

Sales Cloud má několik standardních objektů, jejichž využití Salesforce předpokládá ve všech společnostech. Jsou jimi: Lead (Prodejní tip), Account (Společnost), Contact (Kontakt), Opportunity (Obchodní příležitost). Tyto objekty mají předem definované vztahy a pravidla. Standardní objekty se dále dají upravovat dle přání zákazníka. Zákazníkovi můžeme vydefinovat i novou strukturu aplikace Sales Cloud pomocí tzv. Custom Objects (Vlastní objekty). Custom objekty však postrádají některé výhody přednastavených funkcí standardních objektů. Nelze jich plně využívat v dalších případech jako je např. reporting či zmiňované rozšíření do MS Outlook. Sales Cloud je aplikace, která je předurčena k přizpůsobení zákazníkovi, ale díky daným pravidlům udává obecný obchodní proces, který je dle Salesforce úspěšný a rentabilní. Licenci Sales Cloud lze, stejně jako všechny licence Salesforce aplikací zakoupit pouze přímo od společnosti Salesforce, prostřednictvím jejích obchodníků. Licence se musí zakoupit pro každého uživatele zvlášť. Je tedy potřeba dopředu jasně vytyčit kdo bude mít do Salesforce přístup.

⁴ Microsoft Outlook je e-mailový klient od společnosti Microsoft a je součástí každé instalace MS Office. Obsahuje také kalendář s ním spojené funkce. Pomocí rozšíření dokáže své funkce přizpůsobit dalším aplikacím.

3.1.1.2 Service Cloud

Service Cloud je další z řady produktů společnosti Salesforce. Zaměřuje se především na customer care, tedy péči o zákazníky. Service cloud dokáže propojit komunikační kanály mezi koncovým zákazníkem a společností. Propojením zde autor myslí přímou či nepřímou interakci se zákazníkem. Jako prostředek umí Service Cloud využívat tyto platformy:

- Email
- Telefon, včetně SMS
- Sociální média
- Stránky určené na self-service⁵
- Stránky s online chatovací službou

Součástí Service Cloud je i Field Service, tedy platforma, která pomáhá řídit zaměstnance přímo v terénu. Salesforce Service Cloud je mířen především na společnosti, které mají potřebu infolinky, komunikace s koncovým zákazníkem nebo přímé doručování produktů [11]. Service cloud je v rámci Salesforce zpřístupněný za pomoci licencí, které je třeba pro každého uživatele.

3.1.1.3 Ostatní aplikace

Společnost Salesforce má mnoho dalších aplikací, které nabízí a které mají podobě jako Sales Cloud a Service Cloud své určení. Jmenovitě se například jedná o Marketing Cloud nebo Manufacturing Cloud. Své portfolio se pak snaží rozšiřovat i o externí aplikace jako je Slack⁶ nebo Tableau⁷, které v budoucnosti hodlá v rámci svého portfolia se Salesforce více svázat [14].

⁵ Stránky, které se pomocí například databáze znalostí (knowledge base) nebo oblíbených otázek a odpovědí (FAQs) snaží pomoci uživateli identifikovat a vyřešit problém. Uživatel pak nemusí kontaktovat společnost na přímo.

⁶ Slack – aplikace, která poskytuje komunikační kanály – zprávy a hovory, organizaci týmu, dělení uživatelů do skupin a mnoho dalšího [12]

⁷ Tableau – nástroj na business analýzu, transformaci dat a jejich vizualizaci [13]

3.1.1.4 CPQ

CPQ (Configure, Price, Quote), volně přeloženo – nastavení, cenotvorba, nabídka. Software CPQ vznikl jako nástavba nad produktem Sales Cloud pod záštitou společnosti SteelBrick, kterou na konci roku 2015 koupila společnost Salesforce za 360 milionů dolarů [15]. CPQ se následně začlenilo do portfolia společnosti Salesforce, jako software, který lze zakoupit formou licencí. CPQ nelze koupit jako samostatnou aplikaci. Očekává se, že společnosti již vlastní nebo zakupuje s licencí CPQ zároveň i licenci Sales Cloud aplikace. CPQ licence se taktéž musí zakoupit za každého uživatele zvlášť a platit na měsíční bázi. Není však pravidlem, aby všichni uživatelé aplikace Salesforce Sales Cloud potřebovali licenci CPQ. CPQ je vyhrazeno především pro obchodní zástupce, manažery a uživateli, kteří pracují s objednávkami a fakturací. CPQ je z pohledu Salesforce vnímána jako aplikace, která se do existujícího prostředí Sales Cloud nainstaluje jako balíček obsahující objekty, atributy, automatizované procesy, třídy, Apex kód⁸, záložky, konfigurační prostředí a mnoho dalšího. Není možné aktivně zasahovat do těchto nainstalovaných částí, upravovat kód nebo mazat pole a objekty. Pokud chceme přístup k těmto funkcionalitám umožnit uživateli, použije se tzv. Permission Set⁹, tedy sada povolení, která je speciálně uzpůsobená pro licenční použití. CPQ má přednastavené role, které mají přednastavené přístupy ke konfiguračním objektům či funkcím, které využívá pouze business. Administrátor Salesforce pak snadno vybírá dle role uživatele, jaké možnosti mu zpřístupní. Sady povolení lze libovolně upravovat.

3.2 Salesforce

Salesforce je komplexní nástroj, který je přizpůsoben k možnosti silných úprav a změn od svého zákazníka (společnosti, která si Salesforce zakoupila). Zároveň umí nabídnout značné množství předpřipravených funkcí, jak pro běžný provoz, tak právě pro možnosti úprav celého systému. Autor této práce se pokusí v rámci této kapitoly stručně popsat některé z těchto funkcí a nástrojů. Díky tomu

⁸ [Viz. Kapitola Apex](#)

⁹ [Viz. Kapitola Permission Set – Sada povolení](#)

3.2.1 Standardní objekty

Standardní objekty jsou v Salesforce od prvního spuštění instance aplikace. V dokumentaci[16] je nyní 991 standardních objektů, které tvoří jádro Salesforce. Zákazník je však vystaven pouze nízkým desítkám objektů, které používá, zbytek je pro účely vývoje a funkce systému. V následujících podkapitolách se budeme věnovat standardním objektům, které jsou často využívané zákazníky. Zaměříme se na jejich speciální a jedinečné funkce, které se nedají replikovat na custom objekty a tím si zdůvodníme jejich použití.

3.2.1.1 Lead – Prospekt

V Salesforce je nativně Lead branný jako první kontakt se zákazníkem, první vizitka, první informace o společnosti apod. V mnoha prostředích Salesforce jsou Leady tvořeny pomocí webových formulářů na stránkách společností, které Leady získávají. Tyto záznamy mají pomoc obchodníkům s tím, koho kontaktovat. Leadům se přidává tzv. scoring, tedy skóre (normálně od 1 do 100). Toto skóre definuje, jak moc je pravděpodobné, že by daný Lead mohl dopadnout ve výsledku jako zákazník se smlouvou. Toto skóre se počítá na základě různých údajů. Leady lze tvořit i manuálně nebo si je nahrát hromadně do SF. V rámci Leadů se také řeší problematika přiřazení na konkrétního obchodníka, proto se často sbírají informace o poloze. Aby se Lead nemusel přiřazovat konkrétně, lze ho přiřadit na frontu (Queue) a tím nechat obchodníky z fronty, aby si sami Lead přiřadili a dále s ním pracovali. V Salesforce se dále Lead přetřansformuje na entity Account, Kontakt a případně Opportunita. Konverze provádí z pravidla obchodník a to manuálně. Lze však nechat Leady interaktivně doplňovat zákazníkem a až je jejich potenciál naplněný (skóre 100), mohou se zkonvertovat automaticky.

3.2.1.2 Account – Zákazník

Jedná se o základní entitu celé aplikace Sales Cloud. Jak už je zmíněno, Salesforce obsahuje tzv. Client 360, což je všestranný pohled na zákazníka. Jedná se tedy především o pohled na tuto entitu a další navázané entity. Cílem je mít veškeré informace přehledně dostupné. Další formou jsou tzv. Person Account[17]. Tento typ se musí aktivovat v nastavení a slouží pro identifikaci zákazníků, kteří nejsou

součástí nějaké firmy nebo společnosti, kterou by Salesforce vnímal jako Account (Právní osoba). V praxi jde o jednotlivce (fyzické osoby), kteří obchodují (odebírají produkty, využívají služby) dané společnosti a potřebují evidenci v Salesforce právě jako odběratel nebo zákazník.

3.2.1.3 Contact – Kontakt

Objekt, který reprezentuje kontaktní osobu. Entita kontakt je rozšiřitelná o tzv. AccountContactRelation[18], což je vazební tabulka mezi entitou Account a Kontakt. Je to z důvodu zamezení duplikací na entitě kontakt a možnosti přiřadit jeden kontakt více zákazníkům, tedy vytvoření vazby m:n. Tvorba AccountContactRelation je volitelná v rámci nastavení Salesforce. Pokud je funkce jednou povolena a uživatelé ji začnou využívat, je třeba vědět, že její zakázání s sebou nese následky. Konkrétněji smazání všech nepřímých napojení mezi zákazníkem a kontaktem, což může mít za následek ztrátu informací.

Kontakt má speciální vlastnost ve zmiňovaném pluginu do emailového klienta Outlook. Salesforce z pole odesílatele zprávy vyčte email a vyhledá v entitě kontakt patřičné informace (email). Díky tomu lze ihned v Salesforce identifikovat odesílatele a spojit s dalšími entitami jako je Zákazník, Příležitost, Servisní případ a další. Pokud je aktivní vazba AccountContactRelation, pak Salesforce hledá pouze mezi AccountContactRelation, které jsou definované jako „Direct“, tedy přímá vazba. Salesforce neumožňuje mít více Direct vazeb.

3.2.1.4 Opportunity – Obchodní příležitost

Obchodní příležitost je jednou z nejzásadnějších business entit pro monitoring obchodníků. Zároveň slouží k hlídání a spravování potenciálních obchodů, které mohou uživatelé uzavřít [19]. Příležitosti mají tzv. Opportunity Pipeline¹⁰, také zvanou Path, tedy životní cyklus, cestu.

¹⁰ [Viz. Kapitola - Object pipeline – Životní cyklus objektu](#)

Aby se dala obchodní příležitost kvantifikovat, musí obchodník učinit odhad. Pro tento odhad je v Salesforce připravena funkcionality kolem atributu Amount, tedy objem/množství.

Dalším podstatným atributem je tzv. Profitability, tedy profitabilita, ziskovost. Toto pole se zadává v procentech. Salesforce v rámci své aplikace Sales Cloud obohatil pole o zvláštní chování. Pole je nastavováno pomocí fází v rámci obchodní příležitosti, ale zároveň je přepisovatelné uživatelem.

Kombinací polí Amount a Profitability je Expected Revenue, tedy očekávaný výnos. Hodnota očekávaného výnosu se počítá jako $\text{Amount} * \text{Profitability}$. Tedy pokud obchodník má potenciální obchod se ziskovostí 1mil, ale je pouze v první fázi, kde je nastavená profitabilita na 10 %, pak je v tomto okamžiku počítáno se 100tis. Tento údaj je využíván hlavně v reportingu a přehledech pro hodnocení obchodníků za určité časové období.

Časové limity jsou nedílnou součástí obchodních procesů, proto Salesforce používá standardní pole CloseDate, kde obchodník odhaduje potenciální uzavření příležitosti.

V rámci základních funkcí aplikace Sales Cloud patří i tzv. Opportunity products, tedy vazební objekt mezi obchodní příležitostí a produktovým katalogem. Obchodníci si mohou nativně vytvářet nabídky a cenové kalkulace pro své zákazníky. Opportunity products jsou poměrně omezené. Salesforce pro účely tvorby robustních nabídek, produktových kombinací a návazných procesů využívá nadstavbu CPQ Software (Configure, Price, Quote).

3.2.2 Record Type

Record type neboli Typ záznamu je vlastnost objektů, která umožňuje zásadně změnit jejich pohled a chování. Jedná se o rozdělení jednoho objektu, kde lze definovat jiné uživatelské prostředí, rozlišovat business procesy, mít jiné hodnoty číselníků, filtrovat nebo validovat vstupní data. Vše se pak může na základě přiřazených uživatelských profilů zobrazovat různě.

Je možné rozdělit téměř všechny standardní i custom objekty včetně Account a Opportunity, na více typů. Příkladem může být rozdělení na B2B a B2C klientelu, kde Account i Opportunity budou mít dva record type. Pro každý bude definován jiný

business proces (opportunity pipeline) a podoba uživatelského rozhraní (page layout).

3.2.3 Object pipeline – Životní cyklus objektu

Pipeline tedy průběh nebo také path (cesta) se využívá u objektů, kde se očekává nějaký vývoj situace. Pokud se dá popsat objekt pomocí fází, je pravděpodobné, že se využije object pipeline. Jedná se o grafické zobrazení v rámci Salesforce, které dává uživatelům jasný přehled, v jaké fázi či v jakém stavu se daný záznam nachází. Příkladem může být obchodní příležitost. Uživatel – obchodník může například sledovat pomocí pipeline tyto fáze:

- Nová obchodní příležitost (obchodník u klienta ještě nebyl, ale má možnost, že by se zde mohl povést obchod)
- Probíhající jednání (probíhá jednání s klientem, první oslovení, zjištění požadavků)
- Tvorba nabídky (obchodník tvoří nabídku pro klienta)
- Jednání o smlouvě (nabídka byla přijata, nyní se dojednávají právní ustanovení mezi klientem a společností obchodníka)
- Ukončeno úspěchem (obchod se povedl, klient objednal)
- Ukončeno neúspěchem (klient nemá zájem)

Díky těmto fázím může obchodník či manažer na první pohled jasně vědět, jak probíhá obchod. Zmíněné fáze jsou pouhým příkladem a v rámci Salesforce se dají přizpůsobit. Je to z důvodu různorodosti společností a jejich business procesů. Object pipelines na sebe mohou i navazovat. Například nelze uzavřít příležitost jako úspěšně uzavřenou, pokud nebude uzavřena navázaná nabídka či vytvořený objekt smlouvy. Toto lze nastavit například pomocí validačních pravidel. Pomocí pipeline se přesněji určuje i profitabilita

3.2.4 Validační pravidla

Validační pravidla jsou v Salesforce mocným nástrojem, jak hlídat datovou kvalitu a případně uživatelům zabránit v neočekávaném či nechtěném jednání. Validační pravidla mají za úkol uživatele zastavit a upozornit ho na jeho nekorektní

akci/vstup. Validační pravidla, mají podobně jako Formula fields¹¹ část, kde se specifikuje na základě jednoduchých pravidel. Tato programovatelná část očekává dva možné výstupy – True a False, tedy pravda a nepravda. Pokud výsledkem je pravda, pak se spustí. Při spuštění zobrazují chybovou hlášku, která se nastavuje spolu s pravidlem. Chybová hláška se pak může v rámci GUI prostředí objevit u konkrétního atributu či v horní části obrazovky uprostřed.

3.2.5 Uživatelské Role

Role v Salesforce jsou kopií hierarchie společnosti. Jedná se o interní nastavení. Na základě rolí lze automaticky přiřazovat úkoly (Task), servisní případy (Case), obchodní tipy (Lead) a další. Další funkcí rolí je přístup k datům. Toto se musí nastavit v tzv. Sharing settings[20]. Manažeři z principu mají přístup nejen ke svým datům, ale i datům svých podřízených. Naopak uživatel, který je v hierarchii nejnižší by měl mít přístup pouze k datům, které potřebuje, z pravidla tedy nepotřebuje přístup ke všem datům.

Dalším pozitivem nastavení rolí je jejich automatická adaptace na reporting. Toto se používá v případě přehledů nad daty, které jsou nějakým způsobem segmentovány dle uživatelů. Často se jedná o Klíčové ukazatele výkonnosti – KPI[21], kde Salesforce vedoucím týmů obchodníků zobrazuje data pouze o jejich týmu, tedy dle uživatelských rolí.

3.2.6 Uživatelské Profily

Profily jsou hlavní nástrojem pro přidělování práv uživatelům. Salesforce má předpřipravené profily. Administrátor může tvořit i vlastní profily. Pokud se vytvoří nový profil, nemá téměř žádná práva. Až postupným povolováním se uživatelé mohou zpřístupnit objekty, pole, třídy a další funkcionality ze Salesforce. Většina práv má možnost nastavit tzv. Read Only – Pouze čtení, kde uživatel nemůže měnit data, pouze na ně nahlížet, anebo Read/Write – čtení i zápis. Kde je uživateli umožněno daný objekt, a tedy i jeho pole měnit.

¹¹ [Viz. Kapitola Calculated Field – Kalkulované pole](#)

Speciálním povolením je Read All a Modify All – toto umožňuje ignorovat výše popsané Role a uživatel s těmito právy může číst, případně měnit, veškerá data daného objektu. Tento typ práv se neuděluje příliš často a je určen spíše pro administrátory, developery nebo uživatele z týmu Ops (operations), kteří se starají o správnou funkčnost i datovou kvalitu systému.

3.2.7 Permission Set – Sada povolení

Opakem profilů v rámci Salesforce je tzv. Permission Set, tedy sada povolení. Jedná se téměř o totožné nastavení, které lze provést v rámci tvorby profilů. Rozdíl je v tom, že Permission set používáme pro povolení přístupu a přiřazujeme jej jednotlivým uživatelům. V rámci permission setů se také dají vytvářet tzv. Permission Set Groups, tedy Skupiny sad povolení. Tyto skupiny shlukují jednotlivé sady tak, aby bylo snadné přiřadit více povolení více uživatelům. Vytváříme tak poměrně složitou strukturu, která je však pro korporátní potřeby s tisíci zaměstnanci potřeba.

3.2.8 Speciální pole

Salesforce obsahuje mnoho speciálních polí a datových typů, jejichž cílem je usnadnit práci se Salesforce a poskytnout nadstandartní funkce zákazníkovi bez nutnosti programování.

3.2.8.1 Compound Field – Sloučené pole

Jako příklad složeného pole si můžeme vzít například „BillingAddress“, tedy Fakturační adresu. Pole je přímo na entitě zákazník a jeho datový typ je „address“, což je speciální datový typ pro Salesforce[22]. Tato adresa se pak dá ze Salesforce volat pomocí SOAP API nebo REST API, a to jak ve strukturovaném složeném datovém formátu, tak jako navázaný objekt „Address“. Omezením je však to, že je určeno pouze ke čtení (read-only), nelze tak například upravit přímo dané pole, musíme upravovat jednotlivé prvky daného pole. Podobně funguje i pokud chceme adresu exportovat např. pomocí Data Loader[23]. Toto pole si nelze vyexportovat, musíme vždy exportovat jeho jednotlivé složky. Složky tohoto pole jsou atributy, které bychom na adrese očekávali, jako je ulice, město, stát atd. Není možné přidávat

nebo měnit jednotlivé prvky složeného pole. Sloučené pole address je k dispozici na více objektech, konkrétně Account, Contact, Quote nebo User, tedy Zákazník, Kontakt, Nabídka nebo Uživatel.

3.2.8.2 Geolocation – Geolokace

Dalším Sloučeným polem je i Geolocation, tedy geolokace, která obsahuje pouze dva prvky, latitude a longitude, tedy zeměpisnou šířku a délku. Toto sloučené pole je dostupné pomocí SOAP API, REST API i Apex jak přímo, tak i po jednotlivých prvcích. Zvláštností tohoto pole je, že podporuje i tzv. DISTANCE funkci, tedy možnost po zadání dvou lokací vypočítat vzdálenost. Další funkcí je GEOLOCATION, která v rámci funkce DISTANCE umožňuje zadat lokaci pomocí zeměpisné délky a šířky napřímo. Obě funkce se mohou používat s některými omezeními v SOQL¹²[24]

3.2.8.3 AnyType Field – Víceúčelové pole

Anytype Field dokáže vrátet více typů hodnot:

- String – Textový řetězec
- Date – Datum
- Number – Číselná hodnota
- Boolean – Hodnoty True/False (pravda/nepravda)

Používá se zejména v mapování nebo v rámci záznamů změn (history tracking).

3.2.8.4 Reference Field – Vazební pole

Také se vyskytující jako Lookup Field nebo Master-Detail Field je pole, které odkazuje na jiný objekt. Příkladem je objekt Contact, který je ve vztahu n:1 k objektu Account. Tedy na objektu Contact je vytvořen referencované pole „Account“. Pokud na konkrétním kontaktu (Contact) do tohoto pole vyplní uživatel konkrétního zákazníka (Account), vytvoří mezi nimi vazbu n:1, tedy kontakt má právě jednoho zákazníka a zákazník může mít n kontaktů.

¹² Salesforce Object Query Language – obdobná forma SQL (Structured Query Language) vytvořená pro Salesforce.

Přímo v hodnotě lookup field „Account“ na objektu Contact, bychom v databázi našli ID dané entity Account. V uživatelském rozhraní Salesforce zobrazuje odkaz na daný objekt. Jako text odkazu využívá atribut „Name“, což je automaticky vytvořené pole každého objektu.

Rozdíl mezi Lookup a Master-Detail[25]:

- Lookup – jednoduchá vazba 1:n bez restrikcí a omezení.
- Master-Detail – vazba se závislostí.
 - Pokud někdo smaže objekt, který je v tomto vztahu definován jako Master, budou automaticky smazány všechny objekty, které se definují jako Detail, tedy podřazené objekty.
 - Vlastník záznamů charakterizované jako Detail je vždy stejný jako vlastník Master objektu.
 - Detail objekty dědí pravidla o sdílení a přístupu (security) k objektu
 - Přítomnost referencovaného vztahu na stránce objektu je mandatorní.

3.2.8.5 Calculated Field – Kalkulované pole

V uživatelském prostředí se nazývají Formula fields. Kalkulované pole využívají tzv. Formula, což je algoritmus, který si dokáže převzít hodnoty z jiných polí nebo i jiných objektů. Jedná se o lehce pochopitelný kus kódu. Formula fields mají vlastní syntaxi. Mezi základní funkce patří:

- logické operace AND, OR, NOT, ...
- Textové i číselné operace, ISBLANK, ISNULL, BEGINS, HYPERLINK, LEN, RIGHT, LEFT, SUBSTITUTE, ...
- matematické operace +, -, /, *, MAX, MIN, FLOOR, CEILING, ROUND, SQRT, ABS, ...
- porovnávání =, >=, <=, <>, <, >
- logické podmínky IF, CASE, ...

Formula Field umí i referencovat pole z jiných objektů. Pokud se jedná o odkazování standardního objektu, stačí specifikovat objekt a jméno. Například pokud chceme na

objektu Contact referencovat pole Name na objektu Account, píšeme Account.Name. Samozřejmě se předpokládá, že existuje vazba mezi objekty Account a Contact. Pokud máme však Custom objekty, tedy námi vytvořené, musíme použít syntaxi pomocí referenčního sufixu „r“. Tedy například, pokud máme objekt Account, který má lookup¹³ (tedy vytvořenou vazbu n:1) na objekt My_Custom_Object_c a chceme referencovat například My_Custom_Field_c, což je atribut entity My_Custom_Object_c, pak použijeme My_Custom_Object_r.My_Custom_Field_c. Tento datový typ je velice využívaný nástroj. Jeho výpočet se provádí vždy, pokud je dotazován v právě se vykonávajícím kódu nebo se má zobrazit v uživatelském prostředí.

3.2.8.6 ID Field – Pole ID

Pole ID je tvořené vždy s vytvořením nového objektu. Je to unikátní identifikátor, díky kterému lze referencovat jakýkoliv objekt v Salesforce. Používáme jej při operacích jako je update(), retrieve(), delete(). Salesforce má dva formáty ID – 15 znaková a 18 znaková. 15 znakové ID jsou case-sensitive, tedy rozlišují malá a velká písmena. Nemůžeme tedy říct, že ID 000000000000Abc je to samé jako 000000000000aBC. Aby aplikace jako například Microsoft Office Access¹⁴[26], které nerozlišují mezi velkými a malými písmeny měli možnost identifikovat správná data.

3.2.8.7 Ostatní speciální pole

Salesforce obsahuje další speciální pole[27], která usnadňují programátorům a administrátorům práci a přinášají unikátní funkce do systému. Mezi dalšími je např.:

- ComboBox – Textové pole, které umožňuje uživateli zadat libovolnou hodnotu, ale zároveň obsahuje číselník, ze kterého lze vybírat.
- Currency – Měna – v Salesforce lze nastavit více měnové prostředí. Uživatelé v jedné instanci mohou vidět různé sufixy pro měnu, tedy například „CZK“ za

¹³ [Viz. Kapitola – Reference Field – Vazební pole](#)

¹⁴ Nástroj na správu relačních databází od firmy Microsoft

hodnotu v poli v rámci České republiky a „EUR“ za hodnotou v poli v rámci Slovenské republiky.

- Email – Omezený počet znaků s automatickou validací na podobu emailové adresy.
- EncryptedString – Možnost zašifrování až 175 znaků dlouhého textového řetězce.
- Picklist – číselník, který obsahuje API hodnotu (jednoznačný identifikátor hodnoty) a Popisek, tedy hodnota, která se reálně objevuje v uživatelském rozhraní.
- MultiPicklist – Číselník, který má možnost zvolit více hodnot najednou. Tyto hodnoty jsou odděleny středníkem.
- Percent – Hodnota, která je v dalších výpočtech reprezentovaná jako procento. Tedy pokud máme 50 % ve výpočtu bude figurovat 0.5.

3.2.9 Back End

Back End programátor se zaměřuje na vývoj webové aplikace, tedy všeho, co je mezi webovým prohlížečem a serverem, kde aplikace běží. Back End se v rámci Salesforce chápe jako kód, který se spouští na serverech Salesforce. Tento kód komunikuje s daty, tedy databází a připravuje podklad pro to, co se má uživateli zobrazit. Pokud například uživatel klikne na tlačítko v GUI¹⁵, Front End¹⁶ aplikace komunikuje s kódem aplikace (back end). Na tento popud se spustí příslušný kód a vykoná se daná procedura.

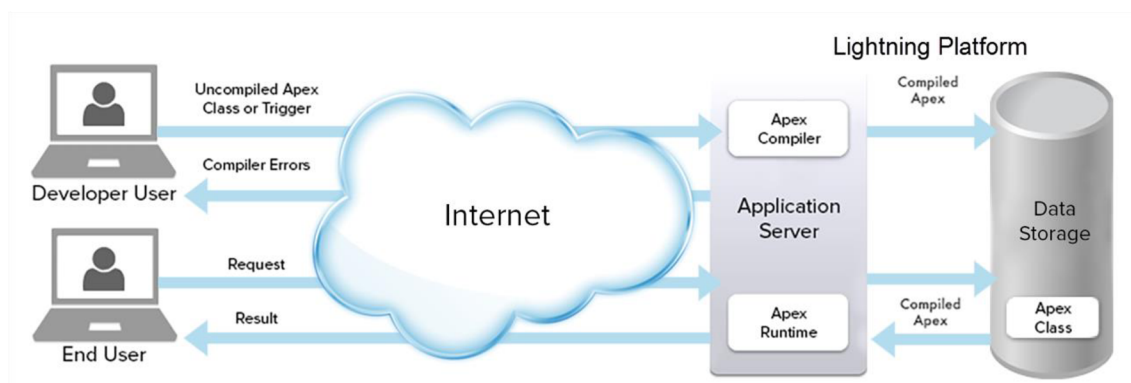
3.2.9.1 Procedurální programování – Apex

Apex je název pro programovací jazyk vyvinutý přímo Salesforce. Jedná se o objektově orientovaný jazyk. Základní myšlenkou objektového přístupu je zapouzdření dat a operací, které je s nimi možné provádět, do jednoho celku, který je nazýván objektem. Tento přístup lze do jisté míry praktikovat přímo v databázové

¹⁵ GUI – Grafické uživatelské rozhraní, tedy vše, co uživatel vidí na obrazovce a s čím může v rámci aplikace pracovat

¹⁶ [Viz. Kapitola – Front End](#)

vrstvě za pomocí jazyka vnořených procedur[28]. Apex je přímo exekuvaný na Salesforce serverech ve spojení s API[29] volání. Syntaxe programovacího jazyka vypadá podobně jako programovací jazyk Java a jeho chování lze popsat jako databázové procedury. Apex dovoluje přidat business logiku k většině systémovým akcím včetně klikem na tlačítko, aktualizace navázaných objektů či Visualforce stránek o kterých se autor zmíní v dalších kapitolách. Apex kód může být iniciován pomocí webových služeb či tzv. trigger na objektech, tedy speciálních tříd, jejichž kód se spouští při různých událostech[30].



Obrázek 1 Chování Apex¹⁷

3.2.9.1.1 Salesforce Object Query Language (SOQL)

Salesforce object query language je jazyk vyvinutý přímo v Salesforce [31]. Je to obdoba příkazu SELECT v známém jazyce SQL¹⁸, tedy Search Query Language, používaný v databázích. SOQL není tak sofistikovaný jako SQL, umí však základní funkce jako např. WHERE – filtrace, ORDER BY – řazení, LIKE – filtrace s nepřesným zadáním, COUNT - počet a další [32]. SOQL se používá při:

- Získání dat z objektu či objektů na sobě navázaných
- Určení počtu záznamů daného objektu
- Řazení výsledku hledání

¹⁷ Obrázek znázorňující chování APEX, převzatý z https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_intro_how_does_apex_work.htm

¹⁸ Deklarativní jazyk pro komunikace a obsluhování relační databáze. Relace představuje souvislost mezi tabulkami relační databáze. OPPEL, Andrew J. SQL bez předchozích znalostí: [průvodce pro samouky]. Brno: Computer Press, 2008. ISBN 978-80-251-1707-1.

- Získání smysluplných dat při používání datových typů jako je číslo, datum nebo checkbox¹⁹

3.2.9.1.2 Salesforce Object Search Language (SOSL)

Salesforce Object search language je jazyk určený ke konstruování textově založeného dotazování proti vyhledávacím indexům. Tento vyhledávací nástroj dokáže dodat maximálně 2000 záznamů, proto je důležité správně nastavit omezující pravidla tak, aby byl výsledný soubor záznamů co nejmenší. Kdy používáme SOSL [33]:

- Získání dat pro specifický výraz, který existuje v rámci dat
- Získání více objektů dle daného výrazu, které spolu nemusí souviset
- Získání hledaného výrazu v čínštině, japonštině, korejštině nebo thajštině

3.2.9.1.3 Integrace s DML (Data Manipulation Language)

V rámci Apex kódu lze psát přímo volání do databáze, jako je INSERT, UPDATE a DELETE, tedy vložení, aktualizace a smazání. Tyto volání mají vestavěnou funkci DmlException, která řeší chyby při operacích v databázi[34].

3.2.9.1.4 Trigger – spouštěcí třída

Apex kód může být spuštěn pomocí tzv. triggers, tedy spouštěcích tříd. Tyto třídy se používají na vykonání kódu před nebo po uložení daného záznamu. Tento kód může obsahovat manipulaci s daty objektu nebo i vytváření, mazání či aktualizace jiných objektů. Apex trigger podporuje spuštění daného kódu po vykonání těchto operací:

- Insert (vložení záznamu do databáze)
- Update (aktualizace záznamu)
- Delete (smazání záznamu)
- Merge (sloučení záznamů)

¹⁹ Pole, které obsahuje hodnoty True nebo False, tedy pravda nebo nepravda. Běžně známý jako Boolean.

- Upsert (vykonání update nebo insert operace, v závislosti na existenci daného záznamu)
- Undelete (obnovení záznamu, v případě, že byl smazán operací delete)

Kód třídy trigger lze psát i nad operacemi standardních objektů, jako je Account²⁰, Contact²¹, Opportunity²² a další.

Existují 2 typy třídy trigger:

- Before trigger – kód tohoto typu třídy trigger se spouští pro aktualizaci nebo validaci záznamu předtím, než je uložen do databáze.
- After trigger – kód tohoto typu třídy trigger se používá v případě, že potřebujeme získat hodnoty, které si ukládá sám Salesforce. Hodnotami může být například Id – identifikátor záznamu nebo atribut LastModifiedDate – datum poslední upravení záznamu. Záznamy v rámci exekuce třídy after trigger je pouze pro čtení [35].

Třídy trigger se mohou navzájem spouštět. Jsou zde však limity, aby nedošlo k přílišnému vytížení Salesforce infrastruktury²³.

3.2.9.1.5 Testování v rámci Apex

Apex obsahuje vestavěnou funkcionalitu pro vytvoření unit testů a jejich exekuci. Unit testy jsou prováděné samými vývojáři. Jedná se o kód, který testuje funkce, procedury či interface aplikace. Cílem je rozložit kód do dílčích kroků, kde se očekává nějaký výsledek. Pokud například funkce obsahuje parametry na vstupu a poskytuje výstupní hodnoty, pak se na tuto funkci implementuje unit test. Tento test se pak při každém nasazení aplikace může strojově spustit a díky zaevidovaným očekávaným výsledkům se vývojář může ujistit, že funkce funguje požadovaným způsobem [36].

²⁰ [Viz. Kapitola Account – Zákazník](#)

²¹ [Viz. Kapitola Contact – Kontakt](#)

²² [Viz. Kapitola Opportunity – Obchodní příležitost](#)

²³ Více o limitech v kapitole [Limity a omezení v rámci Apex](#)

3.2.9.2 Deklarativní přístupy programování v Salesforce

V Salesforce prostředí se deklarativní programování odlišuje od procedurálního programování tak, že namísto psaní kódu uživatel používá předem připravené funkční bloky. Tyto bloky v podstatě představují metody, které vykonávají specifické funkce. Bloky se dále propojují, aby bylo jasné pořadí. Lze i vytvářet podmínkové bloky, které mohou logiku dále větvit. Deklarativní programování v Salesforce je omezené, a to především možnostmi daných funkčních bloků, omezenému logování a následnému trasování případných chyb. Pokud narazíme na omezení, je možné v některých případech „vypomoci“ přímo zavoláním Apex²⁴ kódu (metody). Salesforce má dokonce několik možností, jak jednoduše a bez znalosti programovacího jazyka zautomatizovat procesy. V následujících podkapitolách si je popíšeme.

3.2.9.2.1 Process Builder

Process builder, volně přeloženo jako systém automatizace procesů se používá ke snadnému vytvoření automatizace v Salesforce. Dnes je již brán jako poměrně zastaralý a je aktivně nahrazován Flow²⁵.

Každý proces se skládá z bloků a každý blok zase z kritéria a akcí. Jako kritérium chápeme podmínku. Jako akci či skupinu akcí chápeme vykonanou automatizaci.

Kritérium požaduje na výstupu pouze dvě možnosti – pravda a nepravda. Kritérium se může skládat z mnoha podmínek, které mohou mít svou logiku. K určení logiky vyhodnocení podmínek používáme operátory Booleovské operátory AND a OR[37] případně jejich kombinace. Na základě vyhodnoceného kritéria se větví proces. Větvení procesu není vždy cílem. Můžeme však použít kritérium, jehož výsledkem bude vždy pravda, například identifikátor daného záznamu se nerovná prázdné hodnotě²⁶. Tímto lze přeskočit větvení a spustit požadované akce. Akcí může být více a dělí se na 2 základní typy:

²⁴ [Viz. Kapitola – Procedurální programování – Apex](#)

²⁵ [Viz. Kapitola – Flow](#)

²⁶ Přepsáno do formátu kódu jako Record.Id != null.

- Immediate actions – Akce, které se provedou ihned.
- Scheduled actions – Akce, které budou provedeny v budoucnu. Tento typ akce se nedá navázat na vytvoření záznamu.

Akce v rámci funkcionality Process builder mohou vytvářet nové objekty, aktualizovat data záznamu nebo jeho navazujících záznamů, spouštět další procesy, spouštět Flow²⁷, spouštět metody v Apex²⁸, odesílat notifikace a emaily a mnoho dalšího.

Každý blok procesu končí akcí, která determinuje, zda se má spustit další blok (vyhodnotit další kritérium) nebo se má proces ukončit. Díky tomu není třeba vytvářet více procesů pro daný objekt (záznam), ale stačí správně nastavit kritérium [38].

3.2.9.2.2 Flow

Flow je dalším nástrojem deklarativního programování. Na rozdíl od ostatních deklarativních přístupů, Flow využívá metody podobné procedurálnímu programování. Lze například vytvářet proměnné, smyčky, dotazovat se přímo do databáze či v ní přímo vytvářet, měnit nebo mazat objekty, které jsme si na základě daných kritérií vyfiltrovali. Flow umožňuje uživateli obejít práva na dané objekty, či pole, tedy může být spustitelné v kontextu administrátorských práv. Flow se rozděluje na 2 základní typy:

- Screen Flow – Flow s použitím dialogových oken, které se ve většině případů spouští pomocí tlačítka přímo na stránce v Salesforce
- Autolaunched Flow – Automaticky spouštěné Flow, které se ve většině případů spouští, podobně jako Process Builder²⁹, v momentě změny nebo vytvoření záznamu.

²⁷ [Viz. Kapitola – Flow](#)

²⁸ [Viz. Kapitola – Procedurální programování – Apex](#)

²⁹ [Viz. Kapitola – Process Builder](#)

Screen Flow se používá především, pokud je potřeba vstup od uživatele. Obsahuje prvky, které umožní získat vstup pomocí polí a případně potvrzení pomocí tlačítek. Obrazovky jsou omezené jak do rozložení, tak do funkcionality. Nelze je srovnávat například s tvorbou frontendových komponent, i přes to jsou velice užitečné, jako jednoduchý dialog mezi systémem a uživatelem. Salesforce klade důraz na jejich jednoduchost, pochopitelnost a použitelnost. Díky tomu se však musí upozadit některé pokročilé funkcionality, které se běžně používají v Apex, jako například uložení identifikátoru záznamu, který v předchozím kroku uložen do databáze. I na takovéto funkce existují metody, jak funkcionalitu nahradit, mají však negativní dopady na výpočetní výkon a tím uživatelskou zkušenost (user experience) [39].

3.2.9.2.3 Workflow Rules

Workflow rules, tedy procesní pravidla, jsou jednoduché procesy, které se skládají z pravidla a akce. Pravidla umožňují nastavit 3 možnosti spuštění:

- Při vytvoření záznamu v databázi
- Při vytvoření záznamu v databázi a pokaždé, je-li záznam upraven
- Při vytvoření záznamu v databázi a pokaždé, je-li záznam upraven a splňuje dané kritéria

Kritérium pravidla se podobně jako u Kalkulovaného pole³⁰ určuje pomocí jednoduché syntaxe. Akce Procesního pravidla se mohou aplikovat ihned nebo až po předem určeném čase. Tyto akce jsou omezené a nabízejí například vytvoření úkolu, zaslání emailu, aktualizaci atributu na objektu, v jehož kontextu jsou spuštěny, nebo spuštění jiných akcí v rámci ostatních Workflow rule [40].

3.2.9.3 Limity a omezení v rámci Apex

Apex kód je exekurovaný v multitenant, tedy více-klientském prostředí. Díky tomu musí Apex runtime engine, tedy program v rámci aplikačního serveru Salesforce, striktně limitovat své použití pro jednotlivé koncové uživatele. Pokud Apex kód

³⁰ [Viz. Kapitola – Calculated Field – Kalkulované pole](#)

dosáhne některého z limitů, jeho běh skončí na chybě [41]. Tyto limity se týkají například počtu volání SOQL³¹, počtu získaných záznamů za jedno volání, čas využití CPU³² na Salesforce serverech nebo třeba i počet odesílaných emailů. Vzhledem k možnosti psát Apex kód rekurzivně, existují limity i na tyto případy. Limitů a jejich typů má Apex mnoho, aby nedošlo k vytížení serverů Salesforce případně, aby si nemohly jednotlivé společnosti nárokovat příliš mnoho zdrojů Salesforce platformy.

3.2.10 FrontEnd

FrontEnd je vrstva, která obsahuje veškerý software i hardware, který je součástí User Interface (UI). User Interface je bod, kde se odehrávají interakce a komunikace mezi počítačem (software) a člověkem (uživatel). Součástí UI mohou být jak obrazovka nebo klávesnice, tak vzhled aplikace nebo webové stránky [42]. Lidi nebo jejich digitální reprezentace v podobě uživatelů přímo interagují s prvky ve FrontEnd. Těmito prvky můžeme chápat uživatelem zadávaná data, tlačítka nebo i celé programy a webové stránky. Design těchto prvků pak tvoří User Experience (UX) [43].

3.2.10.1 Visualforce page

Visualforce page je v podstatě framework[44], který umožňuje vytvářet celé stránky v rámci Salesforce tak, aby byl uživatelský zážitek co nejlepší. Jedná se o možnost napsat si custom aplikace přímo v Salesforce v rámci backendu a dále nad aplikací vystavit frontend právě pomocí Visualforce page. Visualforce page používá tagy podobně jako jazyk HTML. Nad Visualforce page lze vyvolat query, tedy dotazy do databáze, případně na podněty stránky do databáze data ukládat[45].

3.2.10.2 Lightning Web Component

Lightning Component je Framework, který využívá standardní jazyky jako je HTML a JavaScript. Využívá faktu, že se jedná o jazyky, které se procesují v rámci prohlížeče uživatele, nikoliv na serveru. Jsou tak velice rychlé a proto se používají především

³¹ [Viz. Kapitola – Salesforce Object Query Language \(SOQL\)](#)

³² CPU – Central Processing Unit – výpočetní jednotka (procesor)

na menší komponenty na stránce, které nejsou jinak v Salesforce standardně k dispozici[46]. Dobrým příkladem je operace nad více objekty v rámci jedné tabulky či hromadná aktualizace dat v rámci potřebné datové struktury.

3.2.10.3 Salesforce a jeho vývoj

Salesforce už od svého vzniku nesouhlasil s robustním vývojem v řádu několika let trvajících aktualizací například z verze 1.2 na 2.0, kde dochází mnohdy k obrovským změnám a problémům. I když má samozřejmě verzovaný release, vydává vždy novou verzi Salesforce, a to třikrát do roka. Verze nepřináší mnohdy na první pohled patrné změny. Tyto změny jsou předem ohlášeny a všichni uživatelé, a především administrátoři, si je mohou vyzkoušet předem. Všechny release se řádně testují a do produkčního Salesforce se dostávají nové funkce vždy po dlouhém působení na tzv. Sandboxech³³. Salesforce po vydání automaticky nainstaluje novou aktualizaci na Sandboxu, kde si ji mohou administrátoři nejen vyzkoušet, ale i zjistit, zda daná verze nezapříčinila chybu v dosavadním systému (nejčastěji v custom kódu). Salesforce garantuje kompatibilitu veškerých aktualizací zpětně[47].

3.3 Problematika migrací

Migrace jsou obecně téma, které je spojeno s téměř každou změnou software. Předně se definuje, jaká data migrovat. Řeší se jejich kvalita, relevantnost, tedy historie dat – zda je záznamy staršího data nutno migrovat nebo stačí například archivace a přenášet se budou pouze data za posledních 3, 5, 10 let. Tématem kolem migrace dat je vždy jejich čistota. Nechceme si při migraci přesunout data, která jsou nekonzistentní, nevalidní, nesmyslná. K tomuto účelu slouží mnohé datové nástroje, především pak ty, co se věnují ETL³⁴. Čištění dat není jen o validacích, ale i o business kvalitě. Záznam, který obsahuje nerelevantní nebo zastaralé informace je zbytečné přenášet do dalších systémů. Data pak zbytečně zabírají místo, které je v nových

³³ Sandbox je v podstatě testovací prostředí Salesforce skrze všechny poskytnuté licence na celém světě. Jde o označení jakéhokoliv prostředí, které není produkční. Salesforce umožňuje mít v rámci organizace mnoho sandboxů.

³⁴ ETL – Extract, Transform, Load. Jedná se o proces, kde se na vstupu extrahují data, dále probíhá jejich transformace a následně se dále odesílají <https://www.ibm.com/cloud/learn/etl>.

systemech často na cloudu, a tedy dražší než data uložená na interních serverech nebo úložištích společnosti.

3.3.1 ETL a Migrační nástroje

ETL neboli Extract, Transform, Load se používá především pro extrahování dat z více zdrojů. Dále slouží k jejich transformaci, tedy sjednocení či vzniku nových, obohacených dat, a následnému nahrání jejich nové podoby do cílového systému za účelem migrací, reportingu, analýzy apod.[48]. V tomto případě se používají nástroje ETL jako Pentaho Data Integration³⁵ nebo Talend³⁶, které se specializují na napojení na různé zdroje dat. Dále se nad těmito daty provedou transformační procesy – obohacení, vyčištění, přeskupení. Následně je tyto nástroje připraví pro reporting či migraci do dalších systémů.

3.3.2 Salesforce a jeho migrační nástroje

Salesforce ze základu obsahuje prostředky pro datovou migraci. Jedná se o poměrně jednoduchý nástroj, který dokáže v rámci průvodce namapovat dané sloupce csv nebo xls souborů na atributy v Salesforce a importovat je. Tento nástroj je určen pro administrátory Salesforce, kteří si takto mohou jednoduše importovat data většího objemu, která nechtějí zadávat ručně. Existují i externí nástroje, jako je aplikace Data Loader nebo plug-in Salesforce Inspector, které se dokáží napojit přímo na API rozhraní databáze Salesforce a přehledně umožnit uživateli importovat data.

Není vždy potřebný pouze import nových dat, ale i aktualizace nebo smazání dat. Všechny výše uvedené možnosti v rámci Salesforce tyto operace umožňují. Jsou zde však výjimky, díky kterým se lépe pracuje s daným nástrojem. Jedna z nich je například v rámci Salesforce Inspector, kde lze data updatovat nejen na základě interních ID Salesforce ale i dalších polí, která nejsou standardní v Salesforce. Jedinou podmínkou je unikátnost těchto atributů.

³⁵ Pentaho Data Integration – nástroj, který se používá pro zachycení, vyřídění a ukládání dat ve formátu, který je srozumitelný a jednoznačný pro cílový systém https://help.hitachivantara.com/Documentation/Pentaho/7.1/OD0/Pentaho_Data_Integration.

³⁶ Talend – ETL nástroj, který poskytuje procesy pro unifikování dat, validace, integrace na ostatní systémy pomocí API a zvyšuje „data trust“ tedy důvěryhodnost dat pro cílové systémy <https://www.talend.com/products/data-fabric/>

3.4 Integrace v rámci Salesforce

Integrace v rámci systému Salesforce je poměrně jednoduchá a připravená funkcionality. V Salesforce se vyskytuje tzv. Object Service API (Application Programming Interface[49]). Jedná se o již připravené API pro každý objekt v Salesforce. API reflektuje i nutnost vyplnit pole či datové typy objektů.

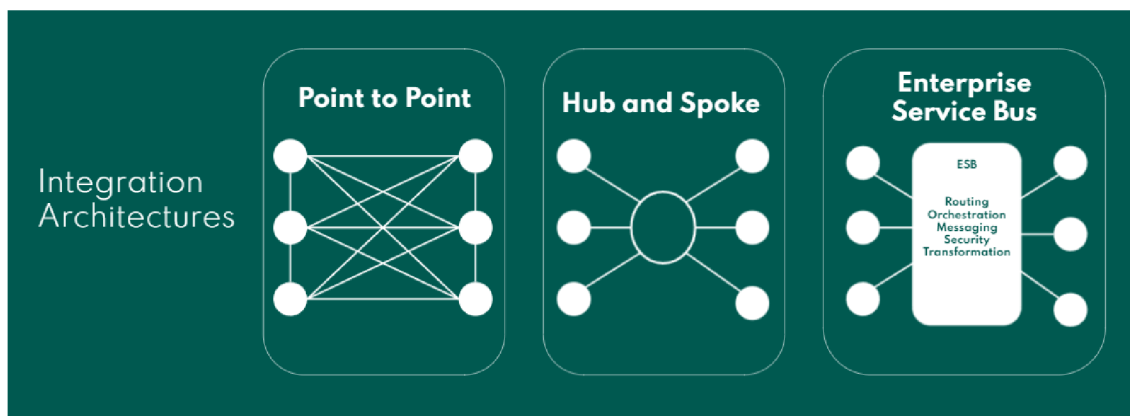
Pokud nestačí standardní API, lze si vytvořit custom API, a to pomocí Apex. Tyto funkcionality využíváme především v případě, mají-li ostatní systémy odlišnou datovou strukturu a není vhodné tyto úpravy dělat na integrační platformě.

Salesforce dokáže komunikovat synchronně i asynchronně. Podporované typy API jsou REST, SOAP, Bulk a Streaming [49].

3.4.1 Architektura Integrací v Salesforce

V Salesforce existují tři typy integrací:

- Point-to-Point – Je integrace, kde Salesforce odesílá zprávy na konkrétní endpoint, tedy API přijímač konkrétní aplikace. Tato integrace je náročná na údržbu, protože v momentu změny je potřeba všechny dané zprávy předělat.
- Hub-and-Spoke – Je integrace, kde Salesforce vše odesílá pouze na jeden endpoint aplikace, která se stará o distribuování zpráv a jejich odpovědí mezi systémy.
- ESB (Enterprise Service Bus) – Je robustní aplikace, která umožňuje routing, orchestraci, transformaci a zabezpečení zpráv (např. WS02). Výhodou je snadná rozšiřitelnost a variabilita použití.



Obrázek 2 Integrovaná architektura³⁷

4 Praktická část

Praktická část diplomové práce se zabývá procesem implementace Salesforce do stávajícího IT prostředí společnosti. Pro potřeby práce bude autor pro tuto firmu používat anonymizovaný termín „Společnost“. Konkrétní prostředí pro realizaci praktické části je inspirováno reálnou zkušeností autora práce. Proces implementace CRM Salesforce, konkrétněji aplikace Sales Cloud spolu s nadstavbou CPQ a integrační platformy WSO2 je v rámci práce rozdělen do 3 fází:

- První fází je popis stávajícího architektonického řešení systémů v dané Společnosti. Také jsou popsány funkce jednotlivých aplikací a jejich vzájemná provázanost. Dále je představena finální vize nového architektonického řešení. Poslední částí první fáze je analýza business procesů včetně jejich výhod a nevýhod pro chod společnosti v době před změnou CRM a **Integrační** platformy.
- Ve druhé fázi je popsána implementace Sales Cloud spolu s WSO2, respektive jejich spuštění do „hybridního režimu“³⁸. Autor popisuje, jaké změny v systému Salesforce a v integrační architektuře musely být provedeny, pro

³⁷ Převzato z <https://www.salesforceben.com/salesforce-integration/>

³⁸ Pojem Hybridní režim se v rámci společnosti ujal, jako popis stavu, kde vedle sebe koexistují dvě různá CRM a propojená dvěma různými integračními platformami. Tento stav není a nebyl nikdy požadovaný a vždy byl vnímán jako dočasný.

úspěšné nasazení fáze 2. Dále je znovu zpracována analýza business procesů včetně nedostatků, problémů, ale i výhod vyplývajících z nasazení fáze 2.

- Ve třetí fázi této práce autor popisuje implementaci zcela nového přístupu. Přichází s řešením, které si dává za úkol omezit či odstranit problémy a nedostatky vzniknuvší ve fázi 2 a zároveň zlepšit celkový stav IT infrastruktury společnosti. Řešením fáze 3 je implementace „mezičlánku“ (nové aplikace) mezi starými a novými systémy společnosti. Autor popisuje vznik nové aplikace – Legacy Adapter, včetně jejího nutného rozšiřování a finální podoby. Dále je popsán důvod, proč byl vybrán právě tento přístup. I když se koncová podoba třetí fáze liší od původně navrženého finálního řešení ve fázi 1, autor popisuje její nutnost pro uskutečnění celkové vize společnosti. V neposlední řadě je opět vyhotovena analýza business procesů.

Na konci praktické části autor zpracoval porovnání všech business procesů ze všech tří fází a vytváří tak pohled na dopady změn systémů z pohledu businessu. Autor zároveň představil řešení, které je použitelné i v podobných případech, jako byl tento. Ono řešení – Legacy Adapter nastiňuje přístup, jak mohou být podobné problémy překlenuty. Díky jeho architektuře má Legacy Adapter možnost rozšíření nebo naopak úplného odstranění s minimálními dopady na zbylou infrastrukturu a ostatní systémy. Jako takový nebrání ve vývoji dalších aplikací, či rozvoje stávajících funkcionalit a je flexibilní při nutnosti přidání nové funkcionality.

4.1 O Společnosti

Již v úvodu jsme zmínili, že se jedná o finanční společnost. Tato Společnost zastává Model B2B, Business To Business – tedy koncový zákazník je právnická osoba. Společnost je střední velikosti a je dceřinou firmou nadnárodní korporace, která má pobočky v dalších zemích Evropy. V České republice měla v roce 2019 cca 100 zaměstnanců. Vlivem pandemie viru Covid19 se počet snížil. Tento krok však neznamenal pokles ekonomického růstu společnosti. Společnost má několik oddělení:

- IT – tým, který pod sebou sdružuje hlavně dodavatele IT řešení pro Společnost.

- DigiOps (Digital Operation) – Tým, který je zodpovědný za chod všech systémů v rámci Společnosti.
- Finance – Tým, který rozhoduje a kontroluje finanční toky ve Společnosti. Má také za úkol reportovat své nadřazené firmě všechny pohyby financí.
- Obchod – Tým, který má rozdělenou Českou republiku na regiony. Každý region má svého manažera obchodu. Tento manažer řídí obchodníky, kteří se snaží prodat produkty dané společnosti firmám ve svém okolí. Tým obchodu má na starosti střední a velké firmy či státní zakázky.
- Telesales – Tým, který má na starosti malé klienty, živnostníky. Obchod probíhá výhradně po telefonu.
- Vedení společnosti

Společnost se rozděluje na 2 oblasti, kde působí.

V první oblasti společnost zprostředkovává užití platebních služeb svým klientům. Říkejme tomu tedy „Klientská oblast/část“. Tyto služby mají možnost odečtu DPH. Společnost v podstatě provozuje platební systém v režimu předplatného (pre-paid). Systém je zprostředkován zaměstnancům klienta. Ti mají možnost v určitých kamenných či internetových obchodech, restauracích, hotelích, sportovních či rekreačních zařízeních finanční služby uplatnit. Těmto obchodům a zařízením můžeme říkat „Partneři Společnosti“.

Druhou oblastí působnosti je následné vyrovnání se s partnery společnosti, kteří si za využití služby nárokují finanční refundaci. Toto vyrovnání je zatíženo marží ze strany Společnosti.

Obchod se tedy realizuje na straně klienta, aby byl co největší odběr finančních služeb. A zároveň na straně partnera, aby bylo možno služby uplatnit.

4.2 Business procesy

V této kapitole budou popsány základní business procesy s jejich obecnou podobou napříč všemi fázemi. Autor použil v následujících kapitolách UML jazyk, konkrétně sekvenční a class diagramy, aby popsal nejdůležitější procesy a vztahy v systémech společnosti. Těmto diagramům je přiřazena časová hodnota na základě poznatků testovacích uživatelů.

4.2.1 Vznik Zákazníka

Entita „Zákazník“ může vzniknout ve více systémech – v Siebel, OMS a v dalších fázích i v Salesforce. Pokud vzniká Zákazník v Siebel nebo v Salesforce, jedná se z největší části o manuální proces. Tento proces je vyvolán uživatelem společnosti (liší se dle fáze). Pokud se jedná o vytvoření zákazníka v OMS, pak se jedná o proces samozasmluvnění. Na vznik zákazníka jsou často navázány i další entity, jako je Kontaktní osoba, Adresa, Bankovní účet nebo i Smlouva.

Aby měl Zákazník pro společnost význam, musí mít dle pravidel společnosti určité náležitosti. Tyto náročnosti zaručují možnost zákazníka kontaktovat či přímo zasmluvnit.

1. Zákazník musí mít vyplněná povinná pole jako je IČO, název zákazníka, DIČ, Typ, Business Segment a další.
2. K Zákazníkovi musí být přiřazena alespoň jedna kontaktní osoba. Tato osoba je považována za kontakt, na který se lze obrátit v případě řešení reklamací, nezaplacených faktur, objednávek a dalších záležitostí. Kontaktních osob může být samozřejmě více.
3. Je třeba vyplnit povinné adresy Zákazníka. Nejdůležitější adresou je právní adresa, která je oficiální pro každou společnost a je uvedená v obchodním rejstříku. Další adresy jsou například fakturační, doručovací či komerční. Pokud tyto adresy nejsou využity, používá se pro vše právní adresa.
4. Bankovní účet je nedílnou součástí dat potřebných pro zasmluvnění zákazníka. Rozlišují se dva typy bankovních účtů – bankovní účet pro prodej a bankovní účet pro vyplácení. Prodejní bankovní účet je pro klienty společnosti. Bankovní účet pro vyplácení se používá zase pro vyplácení partnerům společnosti.

Tyto náležitosti nám reprezentují objekty, které budeme podrobně rozebírat v pozdějších kapitolách.

4.2.1.1 Samozasmluvnění

Samozasmluvnění, je proces, který vzniká na klientském či partnerském portálu společnosti. Tento proces je pro stranu klientů Společnosti zpravidla automatický a nevyžaduje manuální zásah. Pro partnerskou stranu je do dnešního dne bohužel stále s manuálním vstupem. Samozasmluvnění je z pravidla určeno pro malé firmy, které by se jinak nedostaly do hledáčku obchodníků. Tento proces neumožňuje individuální smluvní podmínky, nestandardní požadavky na cenu či provedení produktů společnosti.

4.2.2 Vznik provozovny

Vznik provozovny, tedy určitého místa, kde jsou finanční služby využívány se v průběhu všech tří fází mění. Logika však zůstává stejná. Lze vytvořit provozovnu pouze pokud:

- Existuje zákazník (partner).
- Existuje smlouva na tomto zákazníkovi.
 - Existence smlouvy dále vyžaduje existenci dalších objektů, jako je Bankovní účet nebo Adresa.
- Lze určit o jaký typ provozovny se jedná. Tedy například zda se jedná o květinářství nebo restauraci.
- Lze určit jaké finanční služby se dají na provozovně využívat.

Znalost těchto informací nám umožňuje vytvořit provozovnu, která bude při splnění dalších náležitostí publikovaná na webovém portálu pro zákazníky. Na základě těchto informací a propojením platebních terminálů, lze sledovat platební transakce a vyplácet patřičné částky ponížené o provize společnosti na účty zákazníků, tj. partnerů.

4.3 První fáze

První fáze projektu je v podstatě stav, ve kterém se společnost nacházela před implementací nového řešení. Jedná se o stav, který byl v rámci rostoucí společnosti neúnosný. Byla zde poměrně velká část manuální práce, ruční validace a hlavně –

staré systémy se nedaly příliš snadno rozšiřovat. Nebyla tak možnost držet krok s požadavky byznysu, který zase potřeboval reagovat na měnící se trh. Chyběli administrátoři těchto starých systémů, a pokud už danému systému někdo rozuměl, pak neměl k dispozici dokumentaci a ani ji sám netvořil. I přes to se v reakci na požadavky byznysu nepatrně rozšiřovaly funkce některých novějších systémů. Nakonec se vždy narazilo na limity starých systémů a řešení se nějakým způsobem obešla nejjednodušší cestou nebo se vůbec nerealizovala. Výčet aktuálních systémů první fáze bude následovat v této kapitole.

Společnost byla zároveň jako dceřiná společnost nadnárodní korporace vyzvána vyměnit své stávající CRM řešení za Sales Cloud od Salesforce. To svým způsobem odstartovalo celý projekt nahrazení systému Siebel. Důvodem k výměně stávajícího CRM (Siebel) je především celková globalizace řešení napříč všemi dceřinými společnostmi na celém světě. Cíle globálního řešení:

- CRM (Salesforce) pro každou pobočku.
- Sjedené portály sloužící k nabízení služeb/produktů.
- Bezpečnější IT systémy.
- Reporting napříč celou nadnárodní společností.

4.3.1 Představení starých systémů

V této kapitole autor představí existující systémy společnosti tak, jak existovaly v první fázi. Pokud se jedná o systém, který byl na míru vyvinutý pro společnost, autor přidává prefix „Old“, aby došlo k jednoznačnému odlišení od nově implementovaných systémů.

4.3.1.1 Siebel jako staré CRM

Siebel je CRM systém. Více o CRM systémech jsme si představili v předchozích kapitolách. Siebel je software, který byl založen v roce 1993 a v roce 2002 měl v rámci CRM software zastoupení 45% na trhu [50]. Implementace pro společnost byla upravena na míru, aby vyhovovala poměrně složitému produktovému katalogu. Obchodní proces zůstal až na drobnosti v podstatě podobný jako u všech organizací, které řeší B2B business. Bohužel v rámci této implementace se příliš neomezoval

ani nevalidoval vstup uživatelů. V této fázi projektu bylo běžné, že číselníky obsahovaly stovky položek, data byla nevalidní a některé záznamy byly v nekonzistentním stavu, nemluvě o duplicitách. Implementace Siebelu, která figurovala ve Společnosti, byla častým zdrojem problémů a uživatelům se dostatečně nepředaly informace, jak se systémem správně zacházet. Produktový katalog, který se skládá z jednovrstvé hierarchické struktury, tedy produkt a pod produktem služba, byl naprosto chaotický a nelogický. I přes to byl SSOT – single source of truth, tedy jediným „zdrojem pravdy“ o smluvních podmínkách klienta a veškerých jeho datech. Chyby (především datové), které vznikaly v rámci fungování Společnosti, pak řeší výše zmiňovaný tým DigiOps.

4.3.1.2 Old Invoice System jako fakturační nástroj

Old Invoice System je systém pro tvorbu faktur. Dále slouží jako zdroj pro datový sklad. Jedná se o Oracle databázi, která běží na On-Premise, tedy na serverech patřících společnosti [51]. OIS je také zdrojem pro Data Warehouse – DWH (popsáno níže). OIP funguje pouze pro fakturaci partnerských zákazníků. I když si systém ukládá všechny informace o klientských datech, aby byl jako datový zdroj pro DWH dostatečný, jeho databázové procesy se vztahují pouze na fakturaci pro partnery Společnosti, případně pro fakturaci partnerských provozoven.

4.3.1.3 Old Multifunctional System jako všestranný nástroj

Tento systém původně vznikl jako řešení pro automatizaci objednávek. Dále se rozrostl v portál pro klienty Společnosti, a nakonec převzal i fakturaci celé klientské strany (proto „multifunctional“). Nyní je hlavním nástrojem pro objednávání a správu dat o klientovi přímo pro klienty Společnosti. OMS mimo jiné poskytuje funkci samozasmluvnění – automatizace, kde sám klient vyvolá proces zasmluvnění. Klientovi stačí jen poskytnout potřebné informace o své firmě, kontaktních osobách nebo bankovním účtu. Co se týče smluvních podmínek, klient nemá možnost v tomto procesu dosáhnout nějakých individuálních nebo nestandardních smluvních podmínek. Funkcionalita individuálních smluvních podmínek, byť plánovaná, se do OMS nikdy nedostala. Procesem samozasmluvnění a jeho podobou se budeme zabývat v následujících kapitolách.

4.3.1.4 Old Integration Platform jako hlavní integrační nástroj

Old Integration Platform (OIP) je systém, který se v rámci snazšího pochopení uvádí jako samostatný software. OIP je vlastně pouze součástí Old Multifunctional systému (OMS). Tedy veškeré operace s databází (Oracle DB) či aplikační logika (PHP + Nette) jsou spuštěny na stejných serverech jako OMS. OIP funguje jako hlavní integrační platforma pro 3 důležité systémy – Siebel, OIS a OMS. Jeho hlavní funkcí mimo synchronní i asynchronní integrace v podobě XML souborů je překlad ID. V praxi to vypadá tak, že systém, který je napojený na OIP, pošle v rámci XML zprávy takzvané Caller_ID, tedy identifikátor daného záznamu. Tento údaj je uložen do tabulky IDs pro daný objekt a v rámci synchronní komunikace je zdrojovému systému zaslána zpráva „OK“ či „ERROR“ + text chyby. Následně je XML zpráva s objektem odeslána do ostatních systémů. Tady vzniká asynchronní komunikace, kde OIP v případech založení nového objektu očekává zprávu ID. Ve zprávě ID poskytne obeslaný systém svůj identifikátor daného záznamu, jak si ho uložil do databáze. OIP si tento záznam uloží a pak už komunikuje s daným systémem pomocí jeho interního ID.

Systém tak umožňuje proaktivní monitoring způsobem, jenž dovoluje upravit zprávy, které ostatní systémy odmítly kvůli případným chybám. V praxi to vypadá, že ve frontě odchozích zpráv je možnost editace příchozí zprávy XML. Lze tedy snadno opravit například typ operace – z EDIT na ADD. Toto je využívaná funkce, protože se může stát, že se nepovede odeslat ADD integrační zprávu, ale OIP přijme pouze následnou EDIT zprávu. Tímto nastane nekonzistence dat. Následně pak ostatní systémy zprávu EDIT nezpracují, protože nevlastní daný záznam, na kterém chceme provést EDIT operaci. Zde je možnost, aby administrátor přepsal zprávu z EDIT na ADD a tím založil danou entitu v ostatních systémech. Pokud tak učiní, jakékoliv další zprávy, které čekají ve frontě právě na tuto entitu (navázaná adresa na zákazníkově apod.), budou zpracovány automaticky.

Tento systém je poněkud náchylný na chyby. V momentu, kdy se z nějakého důvodu (např. výpadek systému) nezpracuje zpráva ID, či nastane například chyba v datech, musí se daná zpráva dohledat a integrovat ručně. Na ruční integraci existuje nástroj,

který uživateli umožní zprávu v případě jejího selhání odeslat znovu či dokonce upravit obsah XML zprávy.

4.3.1.5 Data Warehouse jako shromaždiště dat

Data Warehouse neboli datový sklad je typ datového managementu, který se používá především pro analýzu za pomoci nástrojů jako je BI, tedy business intelligence. Obsahuje velké množství historických dat. Tyto data jsou brána ze všech možných systémů, konkrétněji jejich logů, komunikace a podobně [52]. Společnost uchovávala data ze všech systémů a na jejich základě reportovala pomocí software PowerBI³⁹. Hlavním zdrojem dat pro DWH je Siebel a OIS.

4.3.2 Business procesy – první fáze

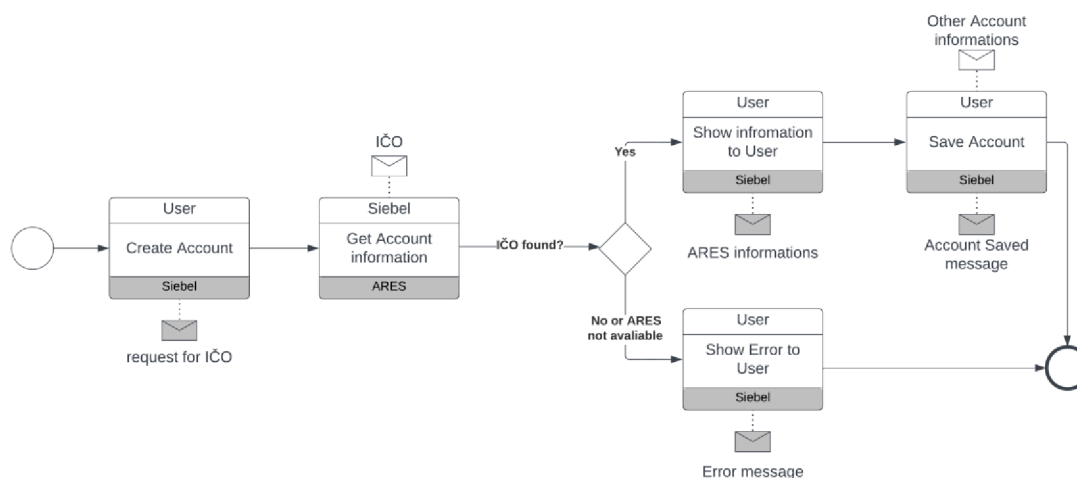
V této kapitole si popíšeme základní business procesy a jejich podobu v první fázi projektu. Autor využije UML jazyk, konkrétně choreografické diagramy, aby popsal nejdůležitější procesy ve společnosti.

4.3.2.1 Vznik zákazníka (první fáze)

Entita Zákazník může vzniknout ve 2 systémech – v Siebel a v OMS. Jak už bylo řečeno, entita zákazník je v rámci první fáze úzce spjata s dalšími entitami, které určují jeho nutnost integrace. Pokud vzniká samotná entita zákazník, pak není důvod pro jeho integraci do dalších systémů. Takto vzniknuvší zákazníci jsou nazýváni Leady/Prospekty a v první fázi jsou vedeni pouze v systému Siebel. Jsou to pouze potenciální zákazníci, které si business oddělení samo zakládá jako datový vstup pro navázání obchodních vztahů. Některé firmy na trhu poskytují seznamy potencionálních zákazníků pro danou firmu, která si od nich seznam kupuje. Tato data jsou pak nahrána automaticky do CRM právě jako Lead/Prospekt, ze kterých následně business čerpá. V rámci první fáze bylo v minulosti těchto Leadu v CRM Siebel vytvořeno několik stovek tisíc. Byla jim doplněna data z veřejných rejstříků, která však nebyla aktualizována. Bohužel tato data v čase ztrácí hodnotu.

³⁹ PowerBI je software pro Business Intelligence, který umožňuje vytvářet reporty, grafy a pohledy na data <https://powerbi.microsoft.com/cs-cz/>.

Pokud vzniká zákazník v Siebel, jedná se z největší části o manuální proces, který je vyvolán uživatelem společnosti. Po vytvoření entity existuje zákazník pouze v systému Siebel, dokud není provedena jeho integrace pomocí systému OIP do dalších systémů. Notifikační zpráva Integrace vyžaduje, aby měl zákazník vyplněné všechna mandatorní pole a navázané další entity – právní a fakturační adresa, bankovní účet, alespoň jednu kontaktní osobu a smlouvu. V Siebel byla vytvořena integrace na veřejný rejstřík ekonomických subjektů – ARES⁴⁰. Siebel po zadání IČO dokázal aktualizovat záznam pomocí dat z těchto veřejných rejstříků.



Obrázek 3 Zákazník z ARES (první fáze) – vytvořeno autorem

Pokud se jedná o vytvoření zákazníka v OMS, pak se jedná o proces samozasmluvnění. Na vznik zákazníka jsou často navázány i další entity, jako je kontaktní osoba, adresa, bankovní účet nebo i smlouva. Daný zákazník se propisuje do Siebel pomocí systému OIP až po kompletním procesu zasmluvnění. Tento proces znamená nutnost mít vyplněná mandatorní data na entitě zákazník a tento zákazník musí obsahovat vyplněné entity adres, kontaktních osob, smluv či bankovních účtů.

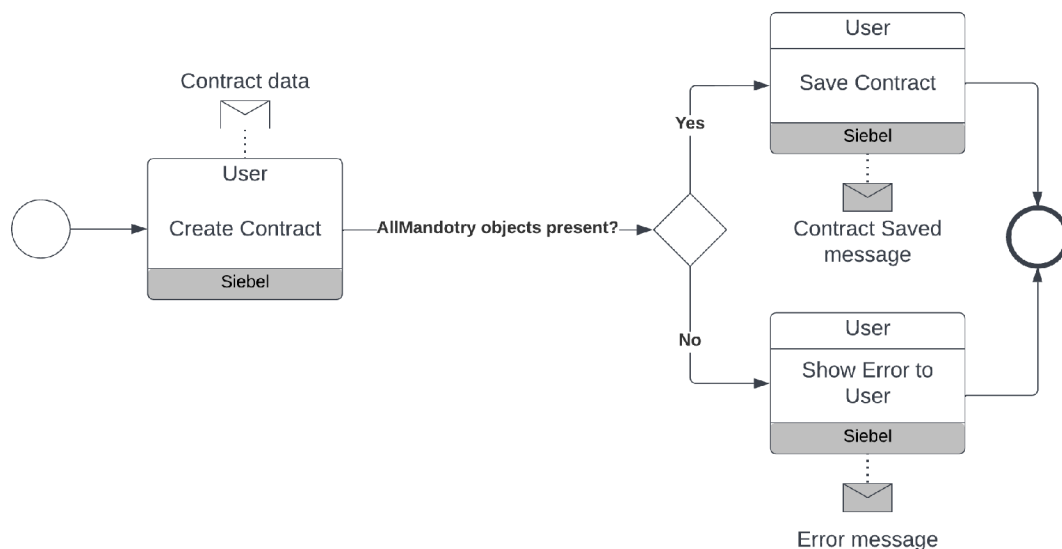
⁴⁰ ARES – veřejný rejstřík ekonomických subjektů ministerstva financí
https://www.info.mfcr.cz/ares/ares_es.html.cz

4.3.2.2 Vznik smlouvy (první fáze)

Vznik smlouvy je v první fázi opět předmětem dvou systémů – Siebel a OMS. Jedná se o entitu, která může a nemusí obsahovat podřazené entity – produkty a služby. Pokud vzniká smlouva v systému Siebel, pak je až na drobné výjimky tvořena manuálně uživatelem, tedy obchodníkem. Smlouva jako taková má své číslo, které je jedinečné a je přiřazováno manuálně. Skupiny čísel smluv jsou v první fázi manuálně rozřazeny na každého obchodníka. Číslo se skládá z čísla pobočky společnosti, kam daný obchodník spadá, posledních dvou číslic kalendářního roku a číselnou řadou, která je generována a rozdělována ručně. Obchodník pak čerpá čísla smluv z tohoto seznamu a postupně je přiřazuje svým novým zasmluvněným zákazníkům. Po zadání potřebných náležitostí na entitu smlouva je před samotnou integrací do ostatních systémů kontrolováno následující:

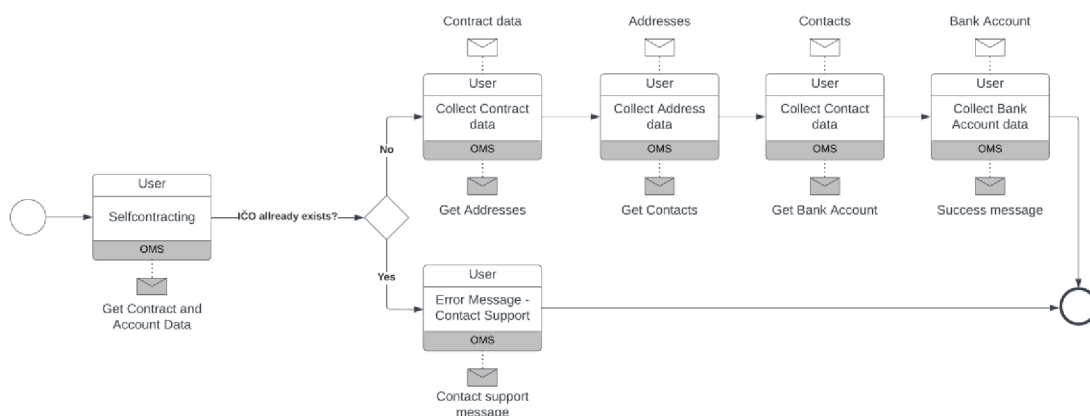
- Existence alespoň jedné kontaktní osoby, která má zodpovědnost za firmu
- Existence právní a adresy doručení
- Existence správného typu bankovního účtu, dle typu smlouvy (klientská/partnerská)

Smlouva, která je tvořena obchodníkem v systému Siebel, může obsahovat individuálního nastavení produktů (INP) smlouvy. INP je entita, která zastupuje jednotlivé produkty a služby na smlouvě v jednovrstvé hierarchické struktuře. Tyto produkty jsou tvořeny v případě, že se nastavení a především ceny liší od standardního ceníku, který je globálně daný pro všechny zákazníky společnosti. Detail INP bude vysvětlen v další kapitole. Existence INP má vysoký vliv na existenci druhé a následně zhotovení třetí fáze projektu.



Obrázek 4 Vytvoření Kontraktu ze Siebel (první fáze) – vytvořeno autorem

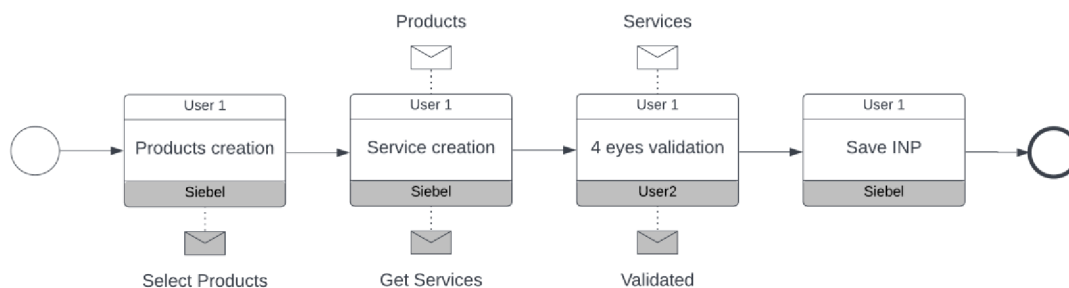
Pokud vzniká smlouva v OMS pomocí procesu samozasmluvnění, je tvořena systémově pomocí vstupu od zákazníka. Číslo smlouvy je generováno automaticky. Podobně jako v systému Siebel jsou v OMS validována data při vstupu do systému a stejně tak při datové integraci na další systémy. OMS při samozasmluvnění nevytváří INP, proto je celý proces značně zjednodušený. Zákazník, který prochází tímto procesem, nemá možnost získat individuální podmínky smlouvy. Jiné ceny než standardní či nadstandardní služby jsou poskytovány pouze pomocí obchodníků Společnosti.



Obrázek 5 Vytvoření Kontraktu z OMS (první fáze) – vytvořeno autorem

4.3.2.3 Individuální nastavení produktů – INP (první fáze)

Nad INP je postavený produktový katalog, který je organizován v jednovrstvé hierarchické struktuře. INP je tedy vazební tabulkou mezi smlouvou a produktovým katalogem. Produktový katalog je klíčovým prvkem pro konkurenceschopnost Společnosti na trhu. Díky vytvořeným INP, tedy produktům a službám pro daného zákazníka, lze zadat výhodnější provizní či pevné poplatky, ale i možnost nadstandartních služeb. V rámci první fáze je produktový katalog značně nepřehledný. Chybí zde pravidla, které by určovala, jaké služby může produkt mít a které již nejsou pro daný produkt možné. Existuje zde spousta produktů a služeb, které se stále dají se smlouvou spojit i když jsou již nevyužívané nebo dokonce společností nepodporované. Neexistuje zde tedy validační logika, která by znemožňovala upozornit koncového uživatele (obchodníka), že danou službu pro daný produkt nelze využít. V rámci Siebel nelze pomocí INP počítat jakékoliv predikce vývoje businessu na daném zákazníkovi. Po schválení, které je popsáno v následující kapitole – Business Model (první fáze), navazuje kontrola čtyřech očí. V procesu kontrolu čtyřech očí musí jiná osoba musí potvrdit správnost zadaných informací. Dále pak obchodník manuálně integruje smlouvu spolu s INP do ostatních systémů. Celkový proces je značně nepřehledný a je v něm prostor pro mnoho chyb. Kvůli tomu jsou už v první fázi data ve špatném stavu.

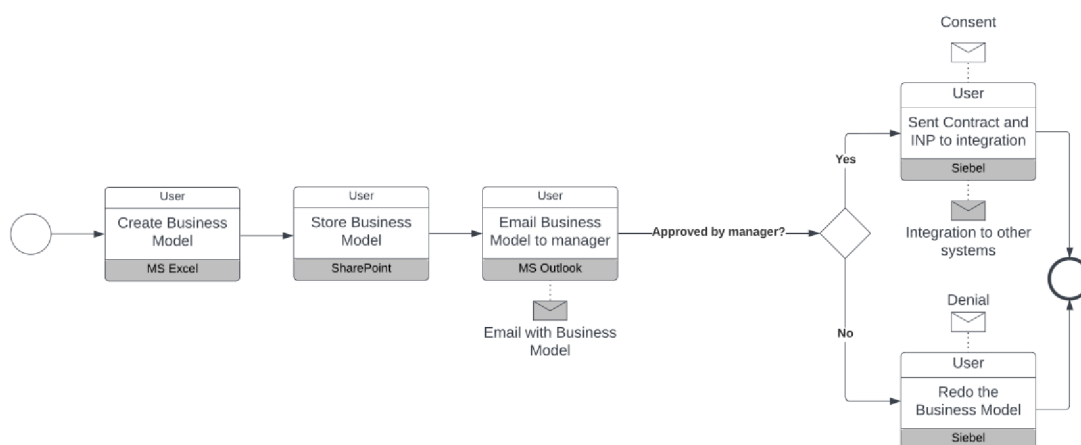


Obrázek 6 Vytvoření INP (první fáze) – vytvořeno autorem

4.3.2.4 Business Model (první fáze)

Business Model je vypočítaný roční obrat, náklady a zisk společnosti pro daného zákazníka, na kterém je Model počítán. I když by se jistě v nějaké verzi CRM Siebel

našel implementovaný Business Model či některá jeho variace, zde tato funkcionality chybí. Tyto všechny pro obchod nezbytné funkce jsou v první fázi exekvované v systému MS Excel⁴¹, kde mají obchodníci předpřipravený základní výpočet Business Modelu. Tento Excelový soubor je potřeba vyplnit ručně na základě dohody se zákazníkem. Další chybějící funkcí, která je nahrazena v této fázi komunikací přes email a sdíleným úložištěm, je schvalování Business Modelu. Součástí Business Modelu je proces schvalování, který je potřebný pro kontrolu nastavení individuálních cen a služeb pro zákazníka. Schválení výše provize či existence některé nadstandartní služby jsou provedeny obchodníkem tak, že se vyplní výše zmiňovaný Business Model. Ten je dále nahrán spolu s dalšími materiály na sdílené úložiště. Následně obchodník napíše email svému nadřízenému, kde žádá o schválení Business Modelu. Pokud nadřízený souhlasí, obchodník vloží potřebná data (INP) do Siebel.



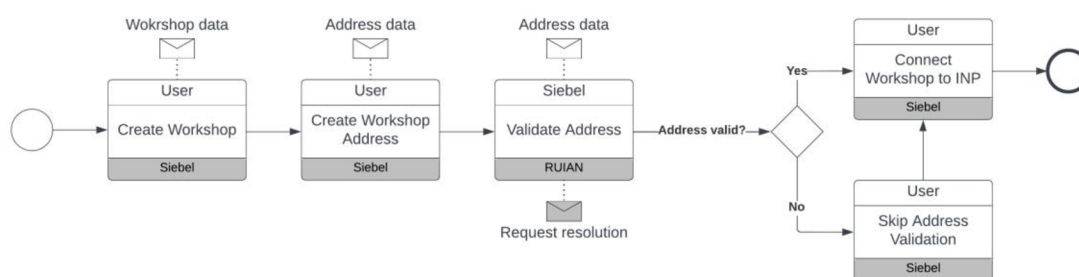
Obrázek 7 Business model (první fáze) – vytvořeno autorem

Vznik provozovny (první fáze)

Pro vznik provozovny je potřeba několik základních údajů. Jedním z nich je jedinečný kód provozovny na zákazníkovi. To je z pravidla trojmístný číselný kód. Bohužel v rámci systému Siebel zde nevzniká validace a provozovny mají tak ve svém kódu znaky abecedy či různé překlepy. Dále se vyplňuje i název provozovny či

⁴¹ Tabulkový systém společnosti Microsoft

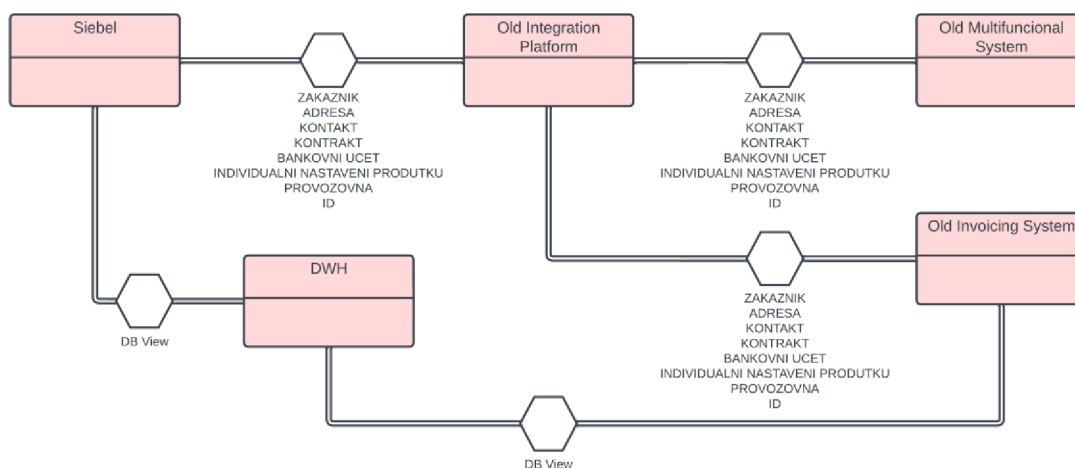
fakturační email. Vznik provozovny je v rámci Siebel provázen zadáním dalších objektů. Těmito objekty jsou adresa a produkt ze smlouvy. Díky adresám lze umístit danou provozovnu na mapě a zobrazit ji koncovým uživatelům (zákazník partnera). O to se pak stará další systém, který pro tuto diplomovou práci není relevantní. Díky přiřazeným produktům na provozovně lze určit jaké produkty daná provozovna přijímá. Jedná se pouze o vztah mezi provozovnou a daným INP. Tento vztah je pak v rámci integrace přenášen přímo na INP. Tím se bude autor zabývat v dalších kapitolách věnovaných integracím. Vznik provozovny je v první fázi možný pouze v systému Siebel.



Obrázek 8 Vytvoření Workshopy (první fáze) – vytvořeno autorem

4.3.3 Integrace systémů – první fáze

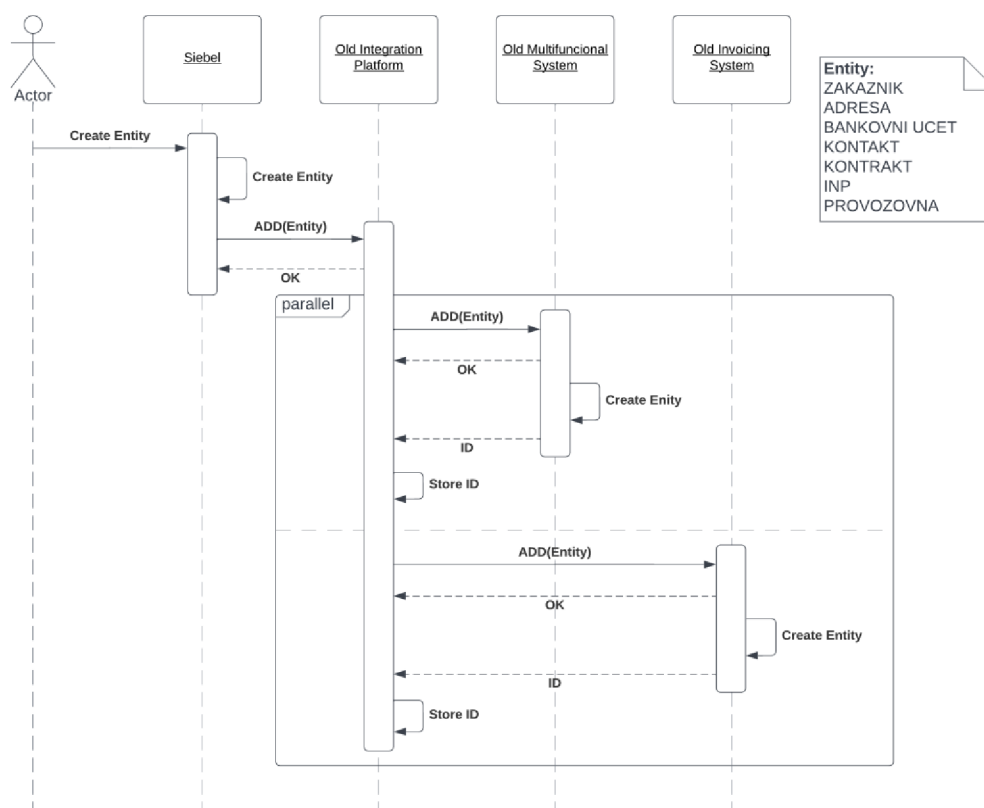
Jak už je zmíněno výše, mezi systémy existuje integrační platforma – OIP. Smyslem je přeposílání zpráv ve formátu XML a překlad identifikátorů jednotlivých systémů. Tento systém má konfiguraci uloženou v tabulkách, kde se určuje, jaký typ zprávy se má odeslat do jakého napojeného systému. V této kapitole se podíváme na to, jak byly systémy propojené v první fázi projektu.



Obrázek 9 Conversation diagram provázanosti systémů v první fázi – vytvořeno autorem

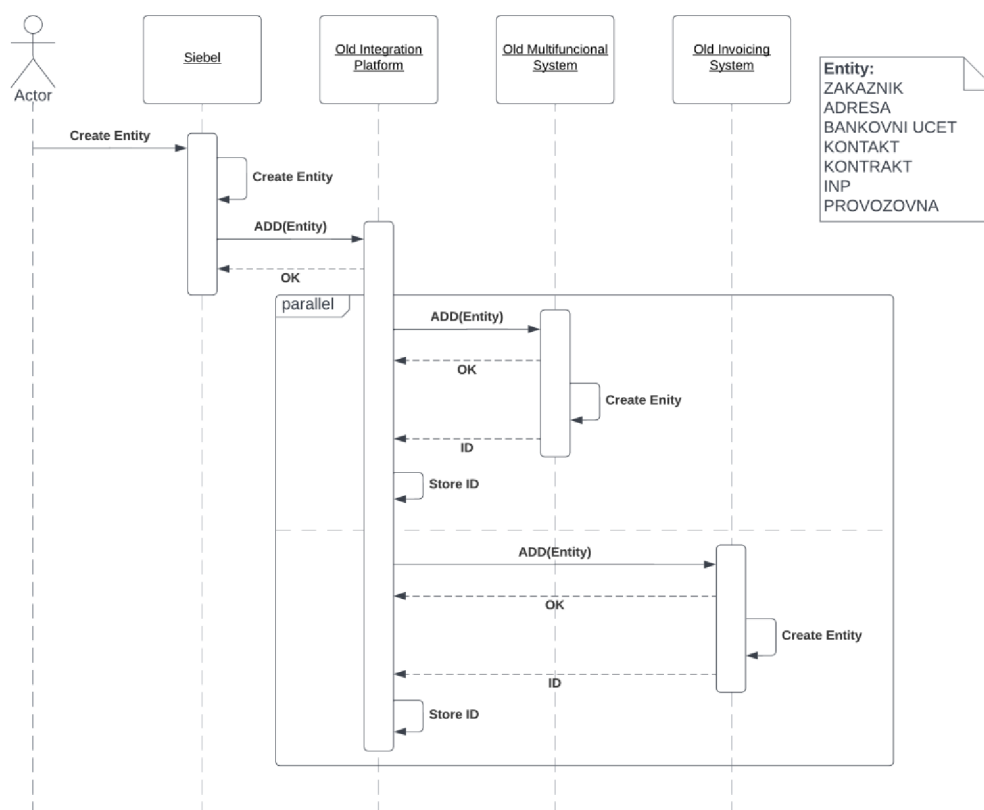
Jak je patrné z obrázku, provázanost mezi systémy je v podstatě přes integrační platformu. Jen do DWH se data dostávají napřímo. Je to z toho důvodu, že DWH si bere data ze systémů přímo z databáze a nepotřebuje tedy validovat vstup, řešit konzistenci dat napříč systémy atd. Data se pouze shromažďují do DWH. Siebel ani OIS z DWH data nečerpají.

V následujících kapitolách si popíšeme, jak funguje tok jednotlivých entit. K tomuto budou sloužit následující sekvenční diagramy. Jelikož se v první fázi neliší způsob toku dat pro všechny entity, byl vytvořen souhrnný digram, který popisuje tok všech entit, a to pro operaci ADD (vytvoření) tak EDIT (úprava) entit. Vzhledem k robustnosti řešení budou popsány entity, které mají v dalším výzkumu smysl a které jsou řízeny CRM systémem. Vynecháme tedy entity jako je objednávka či faktura, které se za všechny 3 fáze v podstatě nezměnily.



Obrázek 10 Integrovaní toky entit při operaci ADD (první fáze) – vytvořeno autorem

Pokud se jedná o vytvoření entity z OMS, je postup naprosto stejný, jako při vytvoření ze systému Siebel. Z OMS však nelze tvořit entitu INP.



Obrázek 11 Integrovaní toky entit při operaci EDIT (první fáze) – vytvořeno autorem

Stejně jako u integračních zpráv ADD, lze všechny entity kromě INP přenést do ostatních systémů (Siebel, OIS) ze systému OMS. Tok i pořadí zpráv jsou identické.

4.3.3.1 Zákazník / Account – ADD/EDIT (první fáze)

Integrace na entitě zákazník je provedena pouze pokud je na ni navázán objekt smlouvy. Tuto integrační logiku zařizuje Siebel případně OMS, pokud jde o samozasmluvnění. Entita zákazník má svojí samostatnou integrační zprávu, a to jak pro vytvoření objektu (add), tak i úpravu (edit) záznamu v ostatních systémech. Masterem, tedy systémem, který je SSOT při případném konfliktu dat, je Siebel i OMS. Toto dvojí masterování záznamu se bude opakovat i pro další entity. Je to z toho důvodu, že oba systémy mají právo pro vyvolání notifikace (add, edit) a tedy možnosti založení/úpravy záznamu v ostatních systémech.

Integrační zpráva pocházející ze Siebel se dostane do OIP. Pokud se jedná o operaci ADD, OIP si vytvoří nový záznam v tabulce IDs a dle konfigurace posílá entitu do

dalších systému. Zpráva v první fázi projektu dojde do OMS a OIS, kde systému odpovídají zprávou OK nebo ERROR. Pokud je vše v pořádku, oba systémy si zakládají záznamy v databázích a po založení odpovídají systému OIP pomocí zprávy ID. Zpráva ID už obsahuje konkrétní identifikátor záznamu z daného systému a OIP si ho ukládá do tabulky IDs, aby mohla při příští zprávě správně přeložit identifikátory. Tímto proces vytvoření zákazníka končí.

Pokud se jedná o operaci EDIT, tedy úpravu záznamu zákazníka, pak je proces zkrácený o návratové zprávy identifikátorů, protože tabulka IDs je již obsahuje. Stačí tedy, aby OIP zařídil překlad jednotlivých identifikátorů. Tedy pokud Siebel pošle XML, kde je Caller_ID záznamu např. „123“, v OMP jako „abc“ a v OIS jako „3f4g5h“, pak OIP zařídí přímo uvnitř zprávy XML překlad tak, že komunikuje na každý systém pomocí jeho vlastního ID.

4.3.3.2 Adresa / Address – ADD/EDIT (první fáze)

Entita Adresa, jakožto podřízený objekt, nemůže existovat bez entity zákazníka. Zákazník, jak již bylo vysvětleno výše, má několik typů adres, jmenovitě právní, fakturační a doručovací. Tyto adresy je potřeba dostat do systémů, které je využívají, tedy do OIS pro fakturaci a OMS pro adresu doručení. Adresa v této fázi projektu je stejně jako zákazník odeslána do těchto systémů přes OIP. Je tu však omezení. Nelze mít více adres stejného typu. Pokud náhodou dojde ke změně adresy zákazníka, adresa doručení se musí „ukončit“, tedy vyplnit datum ukončení adresy. Pokud má klient více poboček, a je tedy potřeba doručovat produkty na více doručovacích míst. Tuto informaci si drží OMS jako více doručovacích adres. Do Siebel je tato informace přenesena jako provozovna (klientská). Jako taková nenesou v Siebel žádnou informaci kromě právě navázané adresy.

4.3.3.3 Bankovní účet / Payment Detail – ADD/EDIT (první fáze)

Bankovní účet je vytvořen vždy v systému Siebel. Dále se integruje přes OIP do OMS, kde je využíván pro párování plateb a objednávek Klientů a do OIP, kde je využíván jako bankovní spojení, kam Společnost vyplácí finance svým Partnerům (partnerská strana).

4.3.3.4 Kontakt / Contact – ADD/EDIT (první fáze)

Kontaktní osoba může vzniknout v OMS i v Siebel. Jejich vzájemná synchronizace je prováděna pomocí OIP. Jedná se o tzv. dvojí masterování entity, tedy oba systémy mají možnost danou entitu změnit a změnu propsat do druhého systému. V případě změny stejné kontaktní osoby v obou systémech najednou se přehrají data změny, která byla zpracovaná systémem OIP jako poslední. Není však velice pravděpodobné, že by zákazník Společnosti a zároveň uživatel Siebelu změnily ve stejný čas tu samou kontaktní osobu.

4.3.3.5 Kontrakt / Contract – ADD/EDIT (první fáze)

Kontrakt je základní entitou pro všechny systémy. Bez tohoto objektu je záznam o zákazníkovi v OMS a OIS nekompletní. Číslo smlouvy jako identifikátor bohužel neslouží na referencování smlouvy napříč systémy. Podobně jako u všech ostatních entit se v OIP ukládá vnitřní ID systému.

Kontrakt vzniká v systému Siebel manuálním zadáním obchodníka nebo backoffice. Smlouva také může vzniknout v systému OMS skrze samozasmluvnění, které je zmiňováno v předchozích kapitolách. Číslo smlouvy má v OMS speciální prefix, aby nedošlo k duplikaci. Kromě čísla smlouvy jsou také velice důležité dva ukazatele, definující, zda je smlouva klientská či partnerská a udávající datum ukončení smlouvy. Datum, kdy smlouva končí, je bohužel v první fázi implementováno poněkud nešťastně. Ve všech systémech (tedy OIP, OMS a v Siebel) je vyplněné datum ukončení smlouvy procesováno tak, že je smlouva ukončená, nehledě na to, zda je datum v budoucnu. Pokud tedy chce uživatel ukončit smlouvu ke konci měsíce, musí ukončit smlouvu ručně poslední den v měsíci vyplněním data ukončení. Toto je cílem v pozdějších fázích napravit.

4.3.3.6 Individuální nastavení produktů / Contract Line – ADD/EDIT (první fáze)

Individuální nastavení produktů neboli INP se integruje samostatně přes OIP jako zbylé entity. Jak už bylo zmíněno výše, jedná se o entitu vazební tabulky mezi kontraktem a produktovým katalogem.

Pro partnerskou stranu je potřebné, aby byla vytvořena všechna individuální nastavení produktů, které chce zákazník (partner) na svých provozovnách využívat. V takovém případě je potřebné vytvořit jak produkty, tak služby a na nich nastavit atribut, který určuje, zda se jedná o individuální či standardní ceník.

Pro klientskou stranu se jedná opravdu o to, co název znázorňuje, tedy **individuální** nastavení produktů. Pokud má tedy klient nějakou slevu ze standardního ceníku, je patřičné pro tuto situaci vytvořit INP. Pokud je však klient zasmluvněn za standardní ceník (typicky samozasmluvnění), pak tvorba INP nemá smysl.

INP má několik speciálních integračních vlastností, které si dopodrobna rozebereme:

- Pokud INP pro daný produkt na klientské smlouvě nefiguruje, znamená to, že daný produkt je za standardní ceník.
- INP v integrační zprávě nese informaci, k jakým subkontraktům je vázaná.

První bod umožňuje, aby smlouva byla platná, ale neobsahovala žádné INP. To však nefunguje pro Partnerskou stranu.

Druhý bod znamená, že místo očekávané vazby, která by se dala integrovat jako samostatná entita, se pouze do XML zprávy INP promítne vazba na jednotlivé subkontrakty. Pokud má tedy partner například 2 provozovny, kde je možnost platit daným produktem, bude mít XML zpráva INP 2 tagy, které v sobě mají informace o vazbě na provozovnu. Celkově pak bude zpráva vypadat např. takto:

```
<?xml version="1.0" encoding="UTF-8"?>
<mw caller_id="INP" entity="INDIVIDUALNI_NASTAVENI_PRODUKTU"
operation="EDIT" caller="SIEBEL">
  <polozka_ceniku>
    <vazba_polozka_ceniku_provozovna>
      <polozka_ceniku_provozovna>
        <IdProvozovna>P-123</IdProvozovna>
        <ProvozovnaActive>Y</ProvozovnaActive>
        <ProvozovnaCode>001</ProvozovnaCode>
        <IdZakaznik>Z-123</IdZakaznik>
      </polozka_ceniku_provozovna>
    <polozka_ceniku_provozovna>
      <IdProvozovna>P-124</IdProvozovna>
      <ProvozovnaActive>Y</ProvozovnaActive>
      <ProvozovnaCode>002</ProvozovnaCode>
    </polozka_ceniku_provozovna>
  </polozka_ceniku>
</mw>
```

```

        <IdZakaznik>Z-123</IdZakaznik>
    </polozka_ceniku_provozovna>
</vazby_polozka_ceniku_provozovna>
    <IdKontrakt>KON-123</IdKontrakt>
    <KontraktType>A</KontraktType>
    <KontraktEndDate></KontraktEndDate>
    <KontraktStartDate>21/08/1968</KontraktStartDate>
    <INPDescription></INPDescription>
    <INPPricingFlag>Y</INPPricingFlag>
    <INPPartNumber>100</INPPartNumber>
    <INParentPartNumber>66</INParentPartNumber>
    <INPActiveFlag>Y</INPActiveFlag>
    <INPPrice>2</INPPrice>
    <INPFrequency>14</INPFrequency>
    <INPPriceIndividual>Y</INPPriceIndividual>
</polozka_ceniku>
</mw>

```

Zdrojový kód 1 – Ukázka XML Inividuální nastavení produktu

Pokud má zákazník (partner) více provozoven, třeba i stovky, pak je XML zpráva patřičně dlouhá a nepřehledná.

Jak už bylo řečeno INP se integruje standardní cestou pro první fázi, tedy přes OIP do OMS a OIS. Pokud vzniká smlouva v OMS, pak se neintegrují žádné INP. Je to z toho důvodu, že při samozasmluvnění nemá OMS možnost jakýmkoliv způsobem upravit ceny produktů nebo aktivnost služeb. Není tedy důvod k vytváření a integrace INP.

4.3.3.7 Provozovna / Workshop – ADD/EDIT (první fáze)

Provozovna se zakládá v systému Siebel a v jistých případech i v systému OMS.

Pokud se jedná o případ, kdy je provozovna založena v OMS, pak jde spíše o založení dalšího doručovacího místa pro klienty. Jak už je popsáno výše v kapitole o integraci adresy, OMS nemá přímo objekt provozovny. V momentu, kdy někdo (zákazník/uživatel) zadá v OMS doručovací místo, do Siebel se přes integrační platformu (OIP) přenesou entita provozovna, která je pouze prázdnou schránkou pro napojení entity adresa, která reprezentuje danou pobočku.

Pokud se provozovna zakládá v Siebel, lze si vybrat, zda bude klientská (poté reprezentuje podobně jako v OMS další doručovací místo) nebo partnerská.

Partnerská provozovna reprezentuje konkrétní provozní místo partnera Společnosti. Při vzniku provozovny se musí určit, jaké produkty je možné na danou provozovnu navázat, jak bylo již popsáno do detailu v kapitole o INP. Tato provozovna má na sobě navázanou i adresu, která reprezentuje místo, kde se provozovna nachází. Dalším a posledním objektem, který je potřeba pro provozovnu vytvořit, je primární zařazení. Tedy určení, v čem daná provozovna podniká (restaurace, hotel, prodejna sportovní obuvi, ...). Jak už bylo popsáno výše, Produkt na provozovně je řešen v rámci integrace pomocí vazby v XML zprávě objektu INP. Adresa se pak přenáší pomocí entity Adresa podobně, jako je tomu u zákazníka. Primární zařazení se nepřenáší přes OIP vůbec. Jedná se o entitu, která je důležitá pro reporting a zobrazení na webovém portálu. Webový portál je oblastí, která není určující pro tuto diplomovou práci.

Klientskou provozovnu tedy lze tvořit v systému OMS i Siebel, ale partnerskou pouze v systému Siebel.

4.4 Druhá fáze

Druhá fáze projektu je výsledek mnoha měsíční práce, kde se od začátku tvořilo na platformě Salesforce – Sales cloud nové CRM, přizpůsobené Společnosti. Jako většina projektů i tento měl termín dokončení, který se nestihl naplnit. Původním návrhem celého robustního projektu bylo jednak nahradit CRM a v ten samý okamžik i nahradit Old Invoicing System a Old Integration Platform. Takto rozsáhlou výměnu však nebylo možná v daném čase stihnout. Staré systémy obsahující roky vývoje od mnoha vývojářů, kde nebyl kladen důraz na dokumentaci. Lidé, kteří něco věděli o daném systému, jeho problematice či o workarounech (provizorních/náhradních řešeních), již ve společnosti nepracují. Během výměny dodavatelů systémů se bohužel nepovedlo vědomosti vývojářů dostatečně předat. V této situaci se nabízí pouze dvě východiska. Můžeme se pokusit vše vyvinout znovu a snažit se pokrýt maximální rozsah starých systémů. Nebo se snažit nějakým způsobem pochopit význam daných systémů a zpětně popsat jejich fungování, napojení na ostatní systémy a dopady napříč celou společností. Velice ambiciózní projekt druhé fáze, která původně měla být finální, se s přiblížením data ukončení projektu ukázal jako nereálný. Po letech vývoje bylo ve všech systémech zaneseno

tolik výjimek, nesrovnalostí, dočasných řešeních trvajících několik let a manuálních zásahů, o kterých měli představu jen vybraní uživatelé, že jeho kompletní nahrazení (ať už se bavíme o Siebel, OIP nebo OIS), byl nereálný. Několik týdnů před plánovaným ukončením vývoje se začala řešit varianta, která by dokázala uspokojit alespoň část z požadavků – tedy nové CRM. Bohužel spousta času a prostředků byla spotřebována pro finální řešení, tedy nahrazení všech 3 systému (Siebel, OIP, OIS) a celkový obraz nového CRM se více či méně odlišoval od původního Siebel CRM. Nakonec bylo rozhodnuto, že veškerou funkcionalitu starého CRM není možná během krátkého času převést do nového CRM a zároveň zachovat kompatibilitu s OMS, OIP, OIS. Bude tedy potřeba paralelní režim dvou CRM zároveň.

4.4.1 Rozdělení starého a nového CRM

Aby došlo k rozdělení entit způsobem, který nejméně ovlivní celý chod systémů a zároveň přinese nové CRM (Salesforce – Sales Cloud), je nejprve potřeba CRM rozdělit na klientskou a partnerskou stranu.

Klientská strana se ve fázi 1 řeší jak v Siebel, tak v OMS. Old Multifunctional System už umí zakládat nové smlouvy (samozasmluvnění) a to bez nutnosti Individuálních Nastavení Produktů. Navíc v rámci vývoje Salesforce proběhlo poměrně rozsáhlé mapování na tento systém. Konexe tedy mezi Salesforce a OMS je daná. Systémy komunikují ze strany Salesforce do OMS a pokud se založí nová smlouva v Salesforce, založí se i v OMS. Tato integrace vznikla za pomoci integračního systému WSO2. Více o WSO2 a jeho uplatnění v rámci fázi 2 bude popsáno v dalších částech. Dále je potřeba, aby se o vzniklém klientovi dozvěděl i systém Siebel a OIS. Tato integrace je již popsána v rámci první fáze, a to v rámci samozasmluvnění. OMS tedy dokáže poslat klienta se všemi entitami:

- Zákazník
- Adresa
- Bankovní účet
- Kontakt
- Kontrakt

Díky tomuto stavu se budou všechny nové klientské smlouvy prezentovat v systému Siebel jako originálně z OMS. Stačí pouze vyřešit otázku vlastníka zákazníka a smlouvy, aby nevystupoval jako systém OMS, ale jako uživatel, který reálně smlouvu založil v Salesforce. Mapování mezi SF a OMS je upraveno a tato problematika vyřešena. Jediným problémem na klientské straně zůstávají INP (Individuální nastavení produktů). Došlo totiž k výraznému předělání produktového katalogu z formy, jaká byla v Siebel, OMS a OIS do nové formy, kterou podporuje Salesforce. Toto bude řešeno v následující kapitole o produktovém katalogu.

Partnerská strana je poněkud obtížnější téma. Jak už bylo zmíněno výše, byl přetvořen produktový katalog tak, aby vyhovoval požadavkům Salesforce. Pokud se však oprostíme na moment od entit související s produktovým katalogem, zbydou nám (podobně jako na klientské straně) entity:

- Zákazník
- Adresa
- Bankovní účet
- Kontakt
- Kontrakt

Zbývají nám tedy dvě základní entity – entita provozovna a INP. INP souvisí jak s provozovnou na partnerské straně (určuje dostupné produkty) tak se smlouvou jako takovou (určuje jaké produkty má reálně partner k dispozici).

Dle celkového rozdělení lze pozorovat, že minimálně 2 entity (provozovna a INP) musí zůstat ve starém CRM – Siebel.

Jak už bylo řečeno, došlo ke změně produktového katalogu způsobem, který by vyžadoval jeden ze 3 možných postupů:

1. Velice náročný vývoj na fakturačním a objednávkovém systému (OIS, OMS), na který nebyl čas ani znalosti daných systémů. Navíc v případě Společnosti se uvažovalo o vypnutí jednoho ze systémů – tedy OIS a další vývoj logicky nedával smysl.
2. Postavení nové aplikace, která by „překládala“ problematiku produktového katalogu a provozoven z nového systému (Salesforce) do toho starého

(Siebel). Na tomto poměrně rozumném kroku by se musela vypracovat detailní analýza o překladu zpráv, mapování, použité technologii na aplikaci a integračním napojení, na kterou nebyl ve fázi 2 čas.

3. Nebo se muselo přistoupit k řešení, které je z obchodního hlediska špatné, ale v tomto případě jediné přijatelné. Toto řešení obsahuje souběh dvou CRM zároveň. Tedy manuální zadávání dat do dvou různých systémů.

Jak je tedy patrné, tak z uvažovaných možností se nakonec vybrala možnost 3, tedy zadávání do dvou různým CRM systémů.

Aby byla komfortnost uživatelů CRM v tomto případě co nejvyšší, rozhodlo se o vytvoření nové integrace, a to ze systému Siebel do Salesforce. Primárním CRM pro zadávání všech entit kromě INP a Provozovna měl tedy být Salesforce. Údaje by však byly v rámci Salesforce neaktuální, proto se rozhodlo o tzv. Rekonciliaci. V rámci Rekonciliace se řešily entity Provozovna a INP tak, že se vytvořil „překlad“ ze starého systému Siebel do podoby Salesforce. Veškeré překlady změn na těchto entitách byly připraveny díky plánované migraci dat mezi systémy. Stačilo tedy „pouze“ tyto migrace naplánovat jako repetitivní proces, který bude do Salesforce dodávat data ze Siebel pravidelně a tím zanechávat ucelený obraz v CRM Salesforce.

4.4.2 Problematika nového produktového katalogu v Salesforce a nemožnost jeho synchronizace 1:1 se Siebel

Změna produktového katalogu byla nevyhnutelnou součástí nového nasazení CRM. Společnost si spolu s licencí Sales Cloud zakoupila i licenci CPQ (problematika CPQ je vysvětlena v teoretické části), která mimo jiné zpracovává pohled a funkcionalitu na produktový katalog. CPQ má za úkol provést obchodníka nastavením nabídky, tedy produktů, služeb a celkové cenotvorby tak, aby nemohlo dojít k nekonzistencím. Zároveň dokáže na proces tvorby nabídky navázat procesem schvalování. Když se vytvářela nová podoba produktového katalogu, bylo jasné, že musí zůstat struktura, tedy produkty a pod nimi služby, jako jednovrstvá hierarchická struktura. Změny v produktovém katalogu byly spíše marginální tedy např. omezení některých kombinací služeb a produktů. Došlo k rozdělení služeb, které byly sloučené a zároveň sloučení služeb, které byly nesmyslně rozdělené. U

některých služeb proběhlo jejich přetvoření na atributy jednotlivých produktů. Největší změnou byl fakt, že INP již nebude reprezentovat pouze **Individuální** natavení produktů, ale přetvoří se na Contract Line⁴². Bude tedy vyjadřovat reálný pohled na to, co klient nebo partner můžou odebírat. Tento zásah nebude však nikdy kompletní, dokud se 100 % neadaptují všechny systémy, které pracují s produkty na smlouvě.

Další velkým rozdílem, ne však nepřekonatelným, jsou tzv. revize smlouvy (verzování smlouvy). V Siebel nebylo možné zjistit původní nastavení smlouvy, protože obchodník nebo backoffice zasahovali přímo do struktury smluvního nastavení (INP) a Siebel si nezaznamenával změny. Jedinou historizací produktů bylo pole, které určovalo, zda je daný produkt ukončený či nikoliv. Nebylo tedy možné zjistit, jak smlouva celkově vypadala v minulé verzi. Toto se mělo změnit s příchodem Salesforce a jeho revizemi, kde bylo možné jasně pozorovat jaké nastavení měl zákazník v minulých verzích a jaké má nyní. Jak ale víme, entita INP se při druhé fázi do Salesforce nedostala a verzování je stále nemožné.

V této kapitole jsme si popsali obecně změny, které se staly součástí release Salesforce. INP se však stále zadává v Siebel, a proto je nutné tyto informace získat do hlavního CRM (Salesforce), aby byl obrázek zákazníka kompletní. K tomu sloužily právě rekonciliace.

4.4.3 Rekonciliace entit aneb zpětná cesta ze Siebel do Salesforce

Návrh rekonciliací byl nasnadě, neboť existoval již vytvořený ETL proces pro migrační logiku, který měl za úkol data ze Siebel přetvořit do podoby, kterou zná Siebel. Proces byl vytvořen za pomoci Pentaho⁴³ od Hitachi Vantara. Díky použití tohoto software na migrace mezi Siebel a Salesforce bylo možné vytvořit i tzv. Rekonciliace, tedy ten samý proces, který se používá v migrační logice, ale je automaticky spouštěn v čase. Toto opakování spouštění procesu způsobilo, že se

⁴² Contract Line použitý název pro objekt Quote Line. Contract Line je v podstatě Quote Line z pohledu integrační logiky, aby bylo přehlednější názvosloví pro okolní systémy. Reálně se však v Salesforce bavíme o Quote Line.

⁴³ Pentaho od Hitachi Vantara – ETL nástroj <https://www.hitachivantara.com/en-us/products/data-management-analytics/lumada-dataops/data-integration-analytics/download-pentaho.html>

data ze Siebel v časových intervalech (cca 1h 30min) synchronizovala ze starého CRM do nového.

4.4.4 Představení nových systémů

V této kapitole si představíme nové systémy, které přibyly v rámci druhé fáze.

4.4.4.1 Salesforce

Salesforce jako nové CRM má ve druhé fázi mnoho funkcionalit. Jak už je popsáno v úvodu kapitoly Druhé fáze, Salesforce přináší masterování mnohých entit, i když z pohledu starých systémů je SSOT stále staré CRM – Siebel. Jako změnu skrze všechny entity přináší Salesforce validace na různá pole, validace business procesů, redukci číselníků, zjednodušení zadávání, validaci v rámci produktového katalogu, graficky příjemnější prostředí, fulltextové vyhledávání napříč všemi objekty a jiné.

4.4.4.2 WS02

Nová integrační platforma, která slouží jako hlavní propojení všech nových a případně i starých aplikací, databází, uložišť apod. Používá při tom API a webové služby jak lokálně, tak v rámci internetu.

4.4.5 Business procesy – druhá fáze

V této kapitole si z pohledu existence nového CRM Salesforce popíšeme znovu hlavní business procesy, které jsme si popisovali v první fázi. CRM Salesforce v rámci paralelního režimu fungovalo spolu se starým CRM – Siebel. Zaměříme se na postupy zpracování dat, které probíhalo i v první fázi a pomocí choreografických diagramů popíšeme jejich chování.

4.4.5.1 Vznik zákazníka (druhá fáze)

Jak už bylo zmíněno, u vytvoření zákazníka v první fázi projektu, existují zákazníci, kteří nemají smlouvu. Těmto zákazníkům se říká Prospekt nebo Lead⁴⁴. Z CRM Siebel bylo migrováno mnoho Prospektů. Vzhledem k informacím, které Prospekty

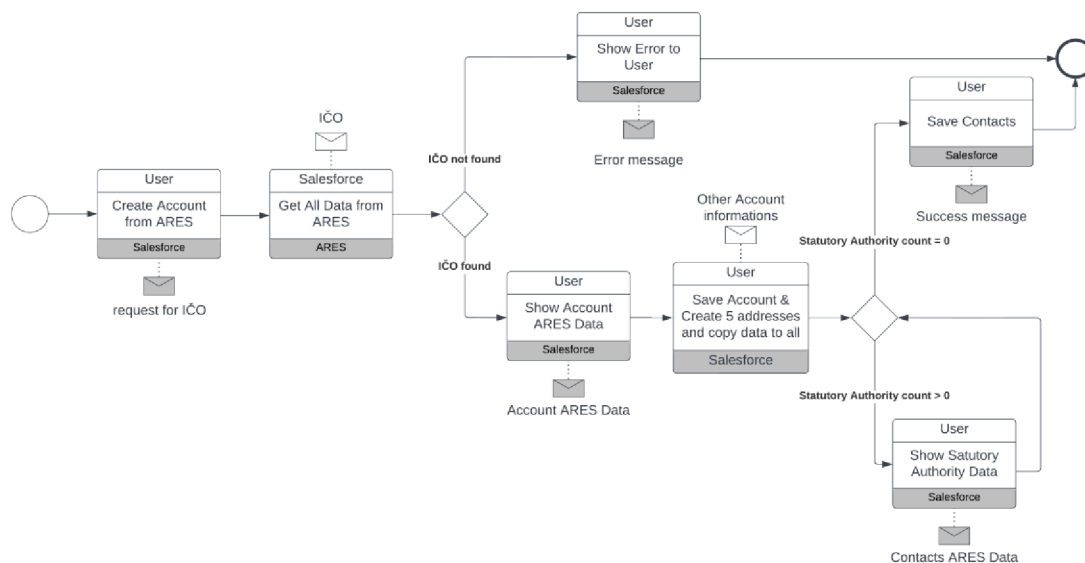
⁴⁴ [Viz. Kapitola – Lead – Prospekt](#)

obsahovaly se vytvořili entity Account typu Prospekt. Tento typ se vyznačuje tím, že nemá žádnou nebo ani jednu aktivní smlouvu. Prospektů byly řádově stovky tisíc, jak už bylo řečeno v první fázi. Na jejich vyčištění nebyl čas.

Pokud obchodník vytvoří entitu Zákazník (Account), jedná se v první fázi o Prospekt. V rámci urychlení zakládání zákazníka je napojen Salesforce na ARES, který dodá informace o dané firmě. V Salesforce bylo implementováno zrychlené vytvoření Accountu pomocí průvodce (flow). Prvním krokem je zadání IČO. IČO je zkontrolováno v rámci databáze Salesforce, aby nedocházelo k duplicitám. Pokud není nalezeno, Salesforce si dohledá informace o firmě v Ekonomickém rejstříku (ARES). Dále se zobrazí přehled doplněných informací z ARES a nevyplněných informací o zákazníkovi. Obchodník nemá možnost přepsat(editovat) informace z ARES, které mimo jiné podléhají kontrole čtyřech očí.

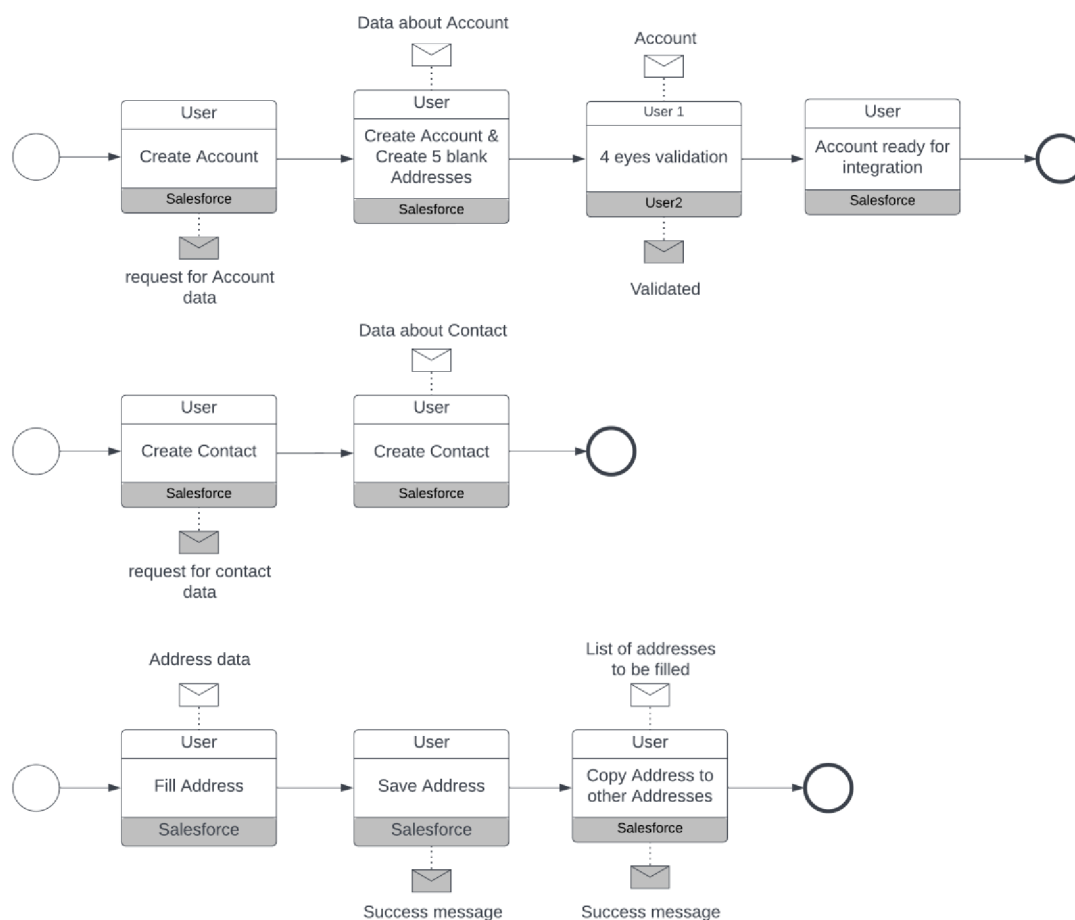
V rámci zakládání zákazníka z ARES dochází i k načtení informací o statutárních orgánech, tedy osobách, které jsou nějakým způsobem v rámci rejstříku zainteresované ve společnosti. Tyto údaje jsou načteny do Salesforce jako Kontaktní osoby.

Dále je v procesu vytvoření zákazníka přes ARES načtena i právní adresa. Tato adresa je založena v Salesforce automaticky a rozkopírována i do dalších typů adres zákazníka. Jedná se o usnadnění pro uživatele, aby nemuseli vyplňovat ručně všechny typy adres. Doručovací adresa sice nemusí být stejná jako právní adresa, ale je rychlejší přepsat jednu adresu než řešit naplnění dalších 4 adres.



Obrázek 12 Zákazník z ARES (druhá fáze) – vytvořeno autorem

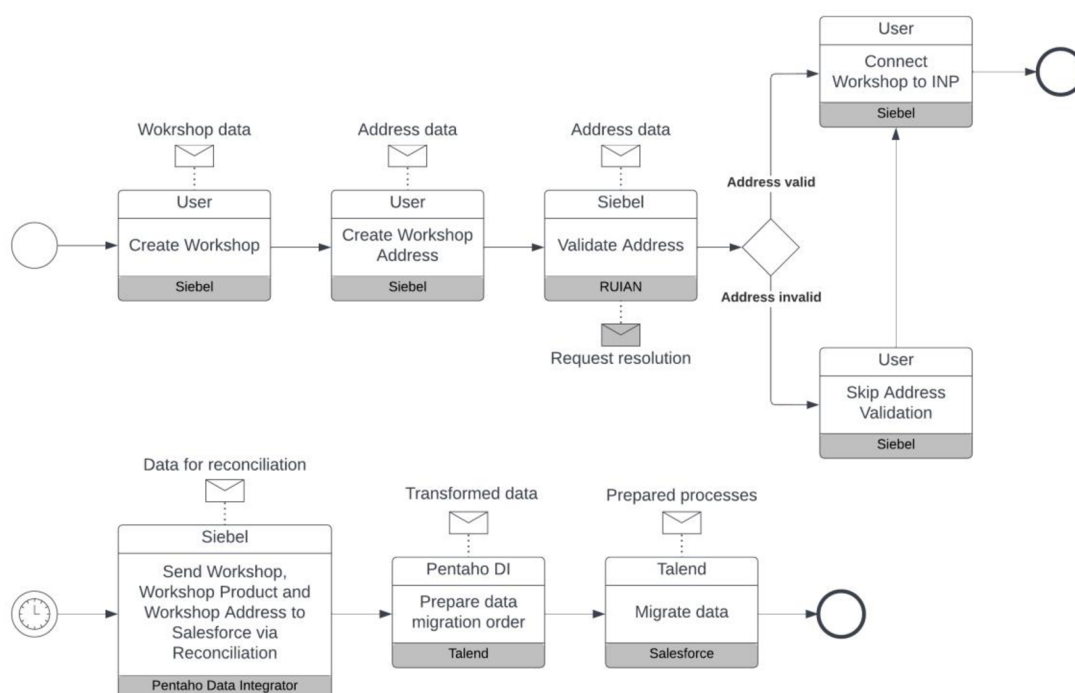
Obchodník také může vytvářet zákazníka manuálně. V tomto případě se spustí klasické vytvoření založení entity Account od Salesforce. Na Accountu bude pro budoucí reference automaticky nastavena informace o tom, že data nejsou importovány z ARES. Aby byl proces srovnatelný s předchozím, kde se zakládá Account, Contact i Adresy z ARES, autor přidal i další diagramy. Na těchto diagramech popisuje nejen vytvoření zákazníka, ale i kontaktní osoby a rozkopírování Adres včetně nutných kontrol 4 očí.



Obrázek 13 Zákazník, adresa a kontaktní osob avytvořena manuálně (druhá fáze) – vytvořeno autorem

4.4.5.2 Vznik Provozovny (druhá fáze)

Provozovna vzniká v rámci druhé fáze stále v systému Siebel spolu s jejími adresami, zařazením a produkty. Nad těmito entitami není žádná validace, kromě toho, že lze přiřadit k provozovně pouze ty produkty, které existují na smlouvě. Zadání provozovny předpokládá, že daný zákazník spolu se smlouvou se prostřednictvím integrací dostal ze Salesforce do Siebel. Aby byly informace kompletní v Salesforce, tak se pomocí reconciliací generují a transformují data ze Siebel. Díky tomu může Siebel sloužit pouze jako místo, kam se potřebná data zadávají a Salesforce jako systém, který všechny CRM data obsahuje a umožňuje jejich náhled.



Obrázek 14 Vytvoření Workshopu, adresy a vazby na INP + rekonciliace do Salesforce (druhá fáze) – vytvořeno autorem

4.4.5.3 Vznik Smlouvy, INP a Business Modelu (druhá fáze)

Abychom popsali správně a detailně celý vznik smlouvy, musíme si tento proces rozdělit do sedmi částí. První částí je vytvoření obchodní příležitosti (Opportunity). Druhou a třetí fází jsou vytvoření nabídky, dále pak vytvoření produktů a služeb, a nakonec určení cenotvorby. Čtvrtou fází je vypočítání Business Modelu. Pátou fází je dále schvalování nabídky. Šestou fází je zasmluvnění nabídky a vytvoření smlouvy. Poslední částí v rámci druhé fáze projektu je rekonciliace do starého CRM – Siebel.

4.4.5.3.1 *Opportunita*

Pro zaznamenání potenciálního obchodu se v Salesforce používá Opportunity. Jedná se o objekt, který se váže přímo na zákazníka a je hlavním nástrojem pro reporting potenciálních zisků Společnosti. Opportunity se rozděluje do 5 respektive 6 fází:

1. Nová příležitost (New Opportunity) – Tato fáze se používá především pro nové zákazníky a je jakým si prvním náznakem toho, že je zde potenciál pro obchod.

2. Objev (Discovery) – V této fázi se už odhadují jednotlivé potenciální zisky. Jedná se o hrubý odhad, který se pak dále v rámci vyjednávání zpřesňuje
3. Vyjednávání (Negotiation) – Do této fáze se obchodník dostane automaticky, pokud je na příležitosti vytvořena nabídka. Jedná o fázi, kdy obchodník nabízí produkty a služby a uzavírá obchod.
4. Uzavřená smlouva (Contract accepted) – Tato fáze se používá v případě, je-li klient zasmluvněn. Podaří se dohodnout na nabízených produktech a jejich cenách a uzavře se smlouva.
5. Uzavřeno úspěchem / Uzavřeno neúspěchem (Close Won / Close Lost) – Do fáze uzavřeno úspěchem se příležitost musí přepnout, v případě, že klient úspěšně objednal. Tímto se považuje celkový obchod za uzavřený. Do fáze uzavřeno neúspěchem lze přepnout příležitost z jakéhokoli stavu. Je to v případě, že klient ztratí zájem.

4.4.5.3.2 Nabídka s produkty a službami

Nabídka – Ve fázi „Objev“ na příležitosti obchodníka je dalším krokem (po odhadnutí zisků) vytvoření nabídky. Nabídka je základním prvkem pro smlouvu. Je nadřazeným objektem pro produkty a služby. V rámci nabídky se vyplňují informace, které pak reprezentují smlouvu. Jedná se o její číslo (nutno vytvořit před vznikem entity smlouva kvůli generování smluvních dokumentů), datum zasmluvnění, osobu podepisující smlouvu, platební metodu a další. Po zadání produktů, služeb a cen se počítá automaticky Business Model. Generují se dokumenty nabídky, smluvní dokumenty, řeší se schvalování a mnoho dalšího. Nabídek v rámci jedné obchodní příležitosti může být více. Pouze jedna je však **primární**. Primární nabídka je ta, která reálně měla úspěch u zákazníka a obchodník ji hodlá zasmluvnit. Nabídky lze klonovat, i když mají vytvořené produkty a služby. Tato vlastnost je atributem aplikace CPQ, která zaštiťuje práci kolem nabídky, produktů, služeb a zasmluvnění.

Nabídka má mnoho fází, které simulují obchodní proces při nabízení produktů zákazníkovi:

1. Rozpracovaná (Draft) – V této fázi se tvoří cenová a produktová nabídka. Do této fáze je možno se vrátit z předchozích fází, aby mohlo dojít k rekonfigurování. Pokud je však nabídka přijata a zasmluvněna, pak návrat do tohoto stavu není možný (validační pravidlo).
2. Ve schvalování (In Approval Process) – Do této fáze se nabídka dostane automaticky po tom, co ji obchodník odešle na schválení. Součástí aplikace CPQ je i schvalovací proces, který je poměrně robustní. Tato funkcionality je však i placenou dodatkovou službou, kterou Společnost nemá. Byl proto implementován vlastní schvalovací proces, který využívá nativní SF funkci – Approval proces. Více o procesu schvalování v této kapitole.
3. Interně schváleno (Sales Approved) – Do této fáze se nabídka přepne také automaticky, a to po schválení od všech uživatelů. Tato fáze má na sobě napojenou notifikaci na email uživatele, který odeslal nabídku do schvalování.
4. Interně zamítnuto (Sales Denied) – Tato fáze se využívá jako indikace, že nabídka neprošla schválením businessu. Pokud v rámci procesu schvalování kdokoliv nabídku zamítne, pak se automaticky přepíná fáze. Stejně jako při schválení i tato fáze má na sobě napojenou notifikaci na email uživatele, který odeslal nabídku do schvalování.
5. Nabídka odeslána (Quote Sent) – Jedná se o fázi, která je nepovinná tedy není potřeba dostat nabídku do této fáze. Slouží jako identifikace pro obchodníka, zda už odeslal nabídku zákazníkovi.
6. Nabídka přijata (Quote Accepted) – Opět jedna z nepovinných fází. Obchodník do této fáze může přejít z fáze nabídka odeslána. Zaznamenává si tak, že tuto konkrétní nabídku zákazník přijal a může se tedy přejít na formulaci smlouvy.
7. Smlouva odeslána (Contract Sent) – Povinná fáze na nabídce, která symbolizuje, že byla smlouva vytvořena a odeslána na podpis zákazníkovi. V tomto okamžiku by měla být na nabídce všechna data potřebná pro zasmluvnění nabídky (hlídáno validačními pravidly).
8. Smlouva přijata (Contract Accepted) – Finální fáze na nabídce, z této je možnost nabídku zasmluvnit a dokončit tak proces.

4.4.5.3.3 Výběr Produktů, služeb a cenotvorba

Produkty a služby se určují ve fázi Draft pomocí CPQ předem vytvořené funkcionality Quote Line Editor. QL Editor má 3 základní kroky – výběr produktů, výběr služeb daných produktů a cenotvorba.

1. Výběr produktů je filtrován podle toho, zda je nabídka klientská nebo partnerská.
2. Výběr služeb je za každý produkt zvlášť. V rámci CPQ je nakonfigurováno, že některé služby jsou povinné a některé nepovinné. Obchodník tak může nadefinovat nabídku přesně podle toho, jaké služby chce danému zákazníkovi nabídnout. Obchodníkovi lze také některé služby nezobrazovat na základě jeho segmentového zařazení. V rámci výběru služeb se také definují atributy pro produkty. Je zde definován odhad počtu odběratelů produktu a frekvence. Tyto údaje jsou nezbytné pro vypočítání Business Modelu.
3. Cenotvorba je posledním krokem pro dokončení nabídky, a určuje je individuální či standardní cena. V rámci cenových pravidel (price rules) CPQ je načten ceník z objektu Price_Table a doplněny standardní ceny (pokud existují). Obchodník nemá vždy možnost ceny změnit – například v případě aktivací služby nemá význam zadávat cenu.

4.4.5.3.4 Business Model

Výpočet Business Modelu probíhá při uložení nabídky v rámci výběru produktů, služeb a cenotvorby. Uložení se spustí QuoteTriggerHandler, což je třída typu Trigger⁴⁵ zodpovědná za veškeré custom operace nad nabídkou. V tomto kódu aplikace se provádí výpočet na základě mnoha parametrů nejen uživatelem zvolených produktů a služeb, ale připočítávají se do něho i náklady. Business Model se počítá defaultně na 3 roky a slouží pro obchodníka jako přehled potenciálního zisku/ztráty pro danou nabídku. Výsledkem Business Modelu je profitabilita, což je číslo uvedené v procentech za dané období (3roky). Tato profitabilita se dále

⁴⁵ [Viz. Kapitola – Trigger – spouštěcí třída](#)

používá do schvalovacího procesu jako prahová hodnota, podle které se nabídka pošle na schvalovací proces. Více o procesu schvalování je v další části této kapitoly

4.4.5.3.5 Schvalování nabídky

Schvalování nabídky je proces, který startuje kliknutím na tlačítko Odeslat ke schválení ve fázi draft a automaticky se vyhodnocuje dle nastavení produktů a služeb. Tento proces má nadefinovaných několik úrovní schvalovatelů. Matice schvalovatelů se upravuje v custom objektu, kde se určují cenové hladiny pro jednotlivé produkty/služby. Tyto hladiny jsou navázané na role schvalovatelů, aby nedošlo k nesprávnému přiřazení na konkrétní uživatele. Fronty uživatelů či jednotliví uživatelé jsou pak definováni přímo ve schvalovacím procesu od Salesforce.

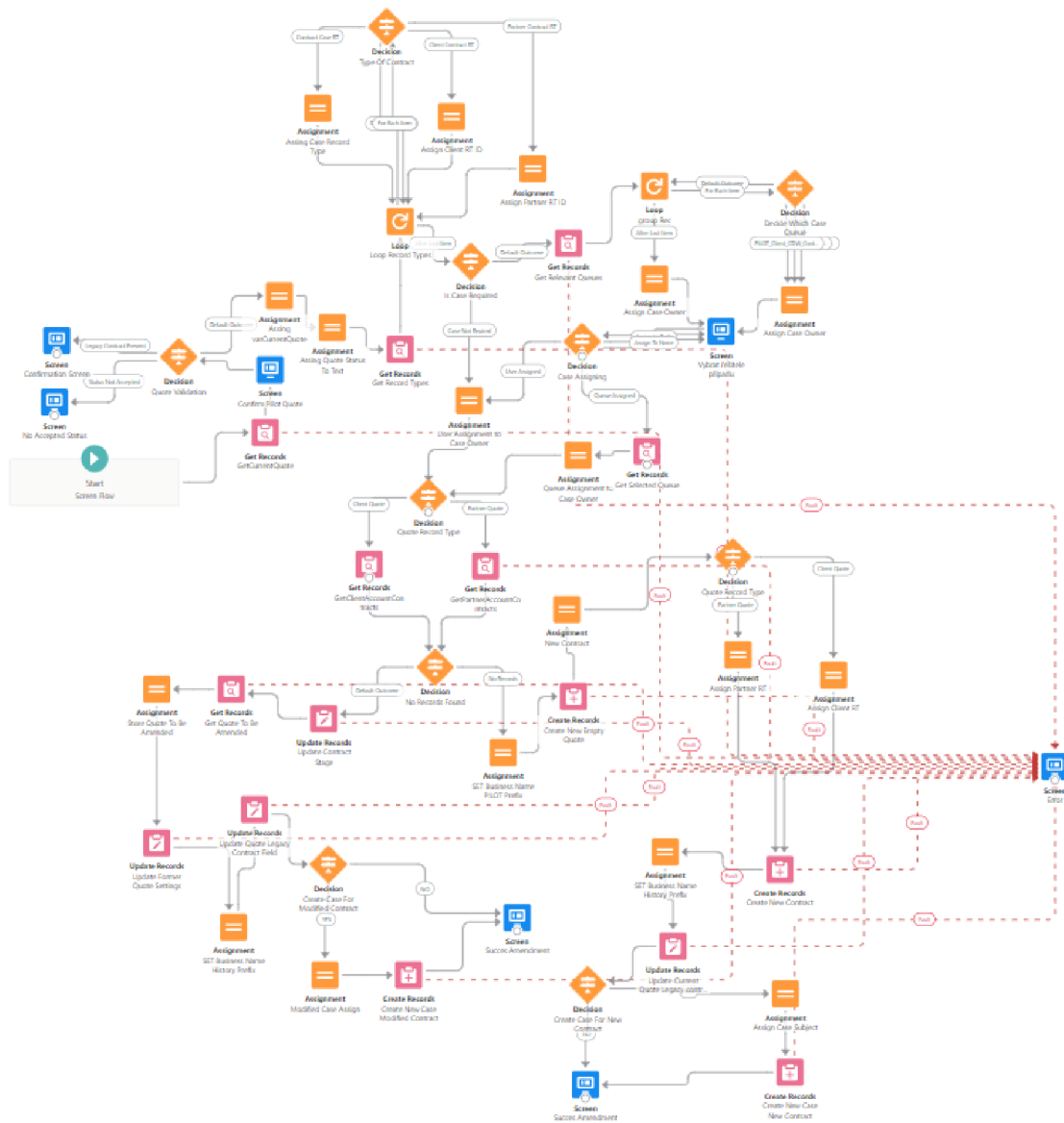
4.4.5.3.6 Zaslavnění nabídky v Salesforce

Pro zaslavnění nabídky v Salesforce nebylo vhodné použít standardní proces, který nabízí CPQ. Důvodů je několik:

- CPQ uvažuje prodej produktů jako kvantitativní záležitost, což Společnost v tomto procesu nevyžaduje.
- CPQ vytváří z Quote Line dva různé objekty – Asset a Subscription. Asset je použit pro jednorázový prodej, tedy pokud prodávám například notebook, jedná se o Asset. Subscription je většinou služba k danému Assetu, v tomto případě by se například jednalo o záruku. Tyto objekty se vytvářejí automaticky dle nastavení Quote Line. V případě společnosti není potřeba tyto objekty vytvářet a plnit tím omezenou kapacitu úložiště.
- V rámci pilotního režimu bylo zaslavnění poměrně nestandardní v tom smyslu, že se musela vytvořit nová – prázdná nabídka, která byla připravena pro naplnění pomocí rekonciliací.

Místo standardního přístupu se vytvořilo Flow⁴⁶, které zajišťovalo potřebné zpracování nabídky ve fázi zaslavnění.

⁴⁶ [Viz. Kapitola – Flow](#)



Obrázek 15 Flow pro zaslumnění (druhá fáze)

Jak je zřejmé z obrázku, flow bylo poměrně komplexní. Proces zaslumnění ve druhé fázi vyžadoval tyto základní kroky:

- Kontrolu, zda nabídka nebyla už zaslumněná
- Získání ID Record Type⁴⁷ nabídky
- Dotaz na uživatele ohledně přiřazení servisního případu (Case).

⁴⁷ Viz. Kapitola – Record Type

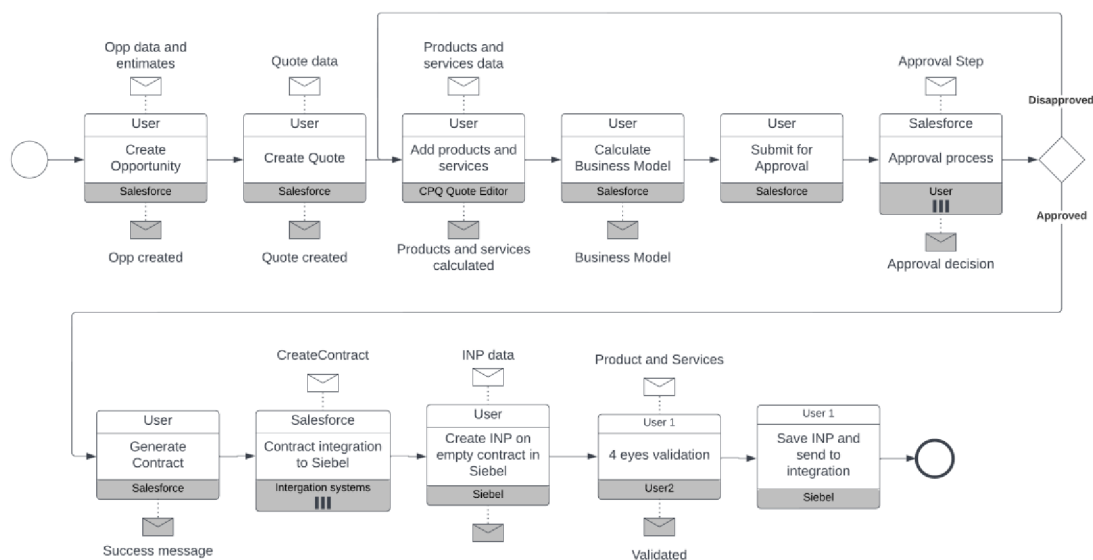
- Získání id fronty/uživatele, na kterého je případ přiřazen
- Zjištění, zda se jedná o novou smlouvy či dodatek smlouvy
- Pokud je smlouva nová
 - Vytvoření nové nabídky (entita Quote), která slouží jako primární pro smlouvu a není již vyměněna
 - Vytvoření nové smlouvy (entita Contract)
 - Přesunutí aktuální nabídky do historických nabídek
 - Vytvoření případu
- Pokud smlouva existuje a řešíme dodatek smlouvy
 - Aktualizace fáze smlouvy na hodnotu „In revision“, která značí, že smlouva není plně aktivovaná
 - Aktualizace údajů v původní nabídce, která byla dříve zasmluvněna
 - Přesunutí aktuální nabídky do historických nabídek
 - Vytvoření případu

V rámci procesu se tedy vytvořila či aktualizovala Quote, která byla pro danou smlouvu primární a ze které se čerpaly veškeré údaje o smlouvě. Pokud byla tato nabídka prázdná (jednalo se novou smlouvu), pak se jedná o nabídku, která byla určená k naplnění v rámci rekonciliací. Pokud se dělal pouze dodatek smlouvy, nové nastavení se uložilo do historických nabídek a primární nabídka na smlouvě „čekala“ na aktualizaci z rekonciliací.

4.4.5.3.7 Dokončení zasmluvňovacího procesu v Siebel

V druhé fázi byl potřeba zásah v systému Siebel, a to pro účely zavedení INP na smlouvu. V rámci zasmluvnění v systému Salesforce se po vytvoření/aktualizaci smlouvy propaly informace skrz OMS a OIP do starého CRM Siebel. Tato integrace trvala řádově několik málo sekund a daný uživatel se tedy mohl přesunout ihned do Siebel a znovu vytvořit potřebné INP. Jedná se o zmiňované dvojí zadávání, které bylo samozřejmě náchylné na chyby. Také zde byl prostor pro úmyslné zneužití faktu že to, co bylo vytvořeno a schváleno v Salesforce, nemuselo být vytvořeno stejným způsobem i v Siebel. Aby k tomuto zneužívání nedocházelo, zůstala nadále kontrola čtyřech očí v Siebel. Po zadání INP se pak v rámci rekonciliačního procesu

přenesla (v řádech hodin) do Salesforce data ze Siebel a naplnila (pokud se jednalo o novou smlouvu) nebo přepsala (pokud se jednalo o dodatek) Quote Line, tedy produkty a služby v Salesforce. Důvodem pro vytvoření nebo existenci právě dvou nabídek (historické a primární) bylo umožnění kontroly, zda uživatel vše zadal správně a zda reconciliační proces funguje, jak bylo zamýšleno.



Obrázek 16 Vznik smlouvy (druhá fáze) – vytvořeno autorem

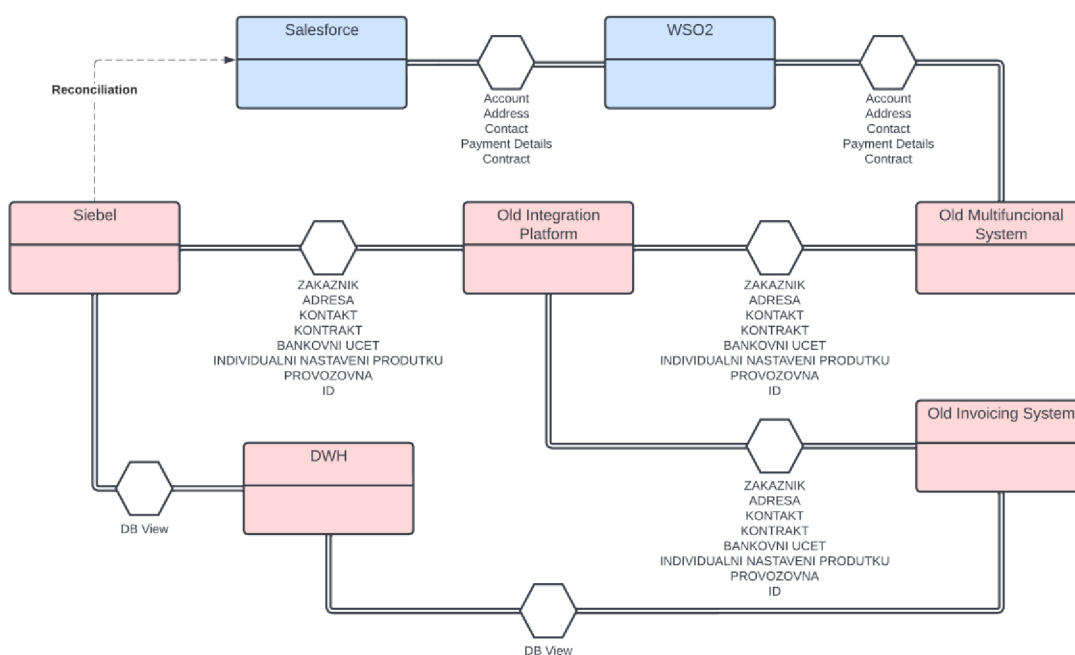
4.4.6 Integrace systémů – druhá fáze

Integrace napříč systémy ve druhé fázi je obohacena o nové CRM, nezbavuje se však žádných starých systémů. Aby bylo napojení v rámci druhé fáze co nejméně náročné na implementaci, zvolila se cesta, která vede primárně přes OMS. Proč bylo zvoleno právě toto řešení, je patrné z následujících důvodů:

- OMS je systém, který v podstatě již umí v rámci starých systémů zakládat smlouvy a přenášet skoro všechny potřebné entity.
- OIP je platforma, která byla určena k vyřazení z provozu. Nemělo tedy cenu se napojovat přímo na ni.
- I když se do Siebel musela dostat všechna potřebná data, aby byly ostatní systémy nezměněné, Siebel by stále systém, který byl primárně určen k vyřazení z provozu. Nemělo tedy smysl se na Siebel napojit napřímo.

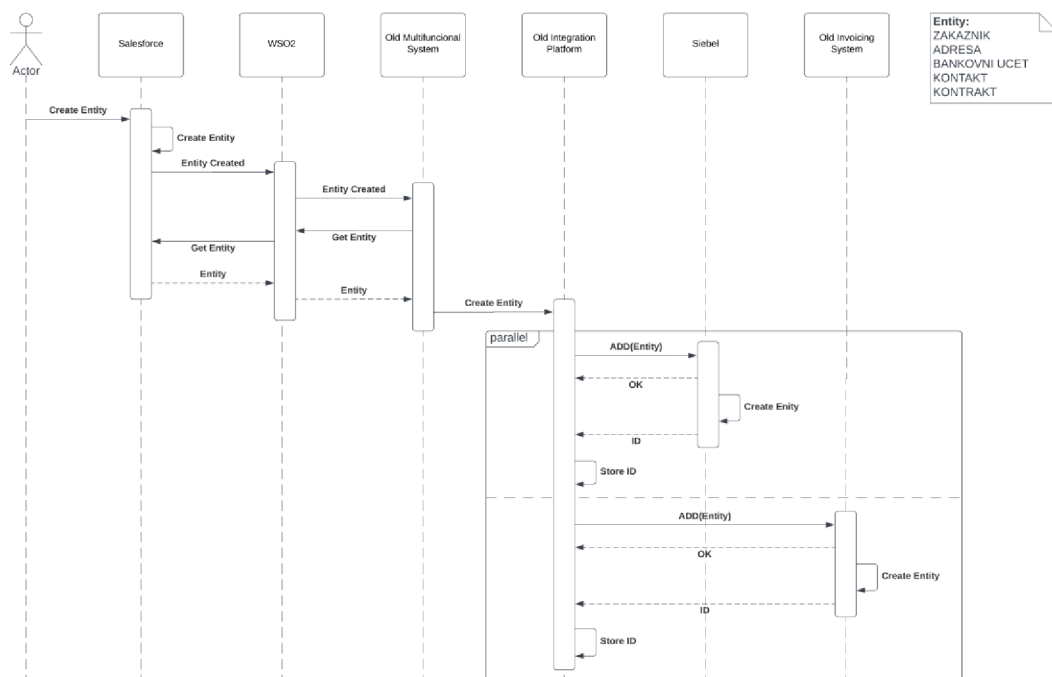
- Termín release Salesforce musel být dodržen a jakékoliv úpravy na starých systémech měly být omezeny na minimum z důvodu končící podpory či nevěli k dalším investicím.
- Místo integrační platformy OIP měla být zprovozněna nová integrační platforma – WSO2, na kterou se v průběhu času měly napojit všechny nové a případně staré systémy (pokud měly být zachovány). Před druhou fází byl již na WSO2 napojen Salesforce, OMS a integrační zprávy se úspěšně přenášely.
- V rámci OMS a Salesforce se počítalo s dvojitým masterováním dat. Tato cesta tedy byla již připravena.

Jak je patrné z obrázku níže, cesta do Siebel ze Salesforce vedla přes WSO2, OMS a OIP. Co se týče rekonziliací, jednalo se o přímou cestu pomocí Pentaho (ETL) a aplikace Talend běžící na migračním serveru.

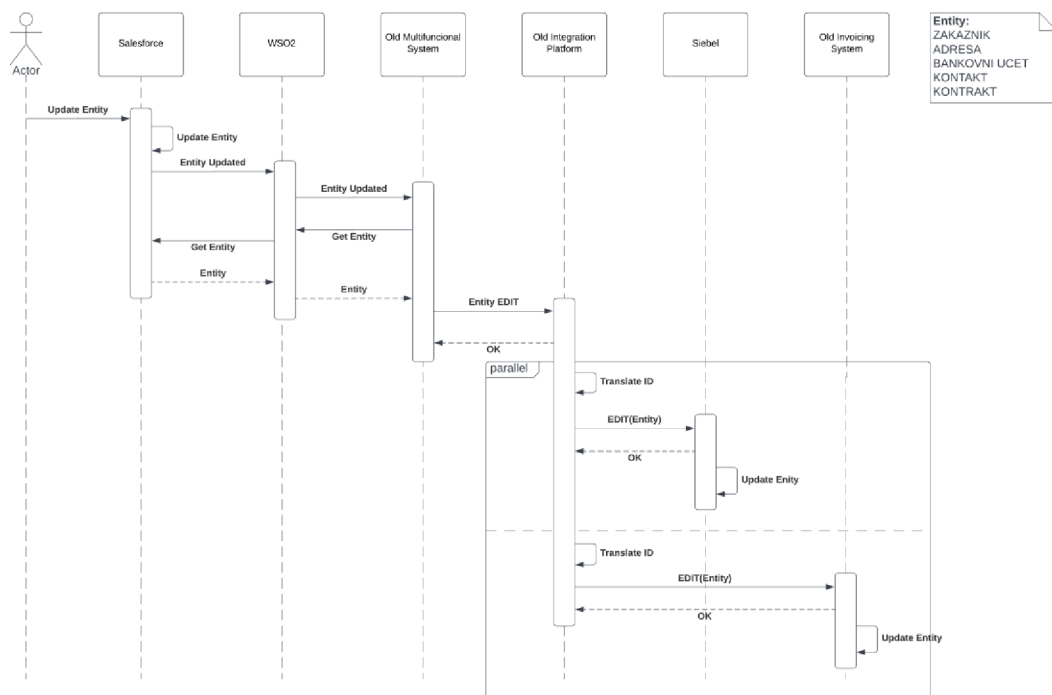


Obrázek 17 Conversation diagram provázanosti systémů ve druhé fázi – vytvořeno autorem

Nyní se zaměříme opět na jednotlivé entity a jejich toky napříč systémy. Na obrázku je viditelné, jaké entity lze integrovat „novou“ cestou. Jedná se o všechny kromě INP a Provozovny.



Obrázek 18 Integrovaní toku entit při operaci ADD (druhá fáze) - vytvořeno autorem



Obrázek 19 Integrovaní toku entit při operaci EDIT (druhá fáze) - vytvořeno autorem

Toky integračních entity Provozovna a INP zůstávají stejné jako v první fázi, pouze se přidává rekonciliace. Rekonciliace tyto entity po jejich vytvoření v Siebel periodicky přenáší do Salesforce, aby vznikl ucelený obraz v hlavním CRM (Salesforce).

4.4.6.1 Zákazník / Account – ADD/EDIT (druhá fáze)

Entita zákazník je nyní masterovaná v Salesforce a OMS, kvůli trvající možnosti samozasmluvnění.

Pokud se zaměříme na manuální vytvoření zákazníka v Salesforce, pak je entita po jejím vytvoření a schválení kontrolou čtyřech očí odeslána na novou integrační platformu – WSO2. WSO2 si vyčte notifikaci o novém zákazníkovi a odesílá notifikaci do předem určených systémů. V tomto případě OMS. OMS má 2 fronty Q1 a Q2. Ve frontě Q1 si ukládá zprávu. Ve frontě Q2 ukládá zpracované zprávy, případně chybové stavy zprávy. Pokud byla zpráva vyčtena korektně, ukládá si OMS zákazníka. Pokud se řeší zákazník v rámci samozasmluvnění, vzniká v OMS. Jeho proces je takový, že po shromáždění všech informací se pošle požadavek z OMS do Salesforce. Vzniknuvší zákazník v OMS není integrován dále do OIP, ale čeká na založení v Salesforce. Pokud se pošle požadavek na samozasmluvnění, Salesforce zpracuje tuto zprávu a založí všechny potřebné entity – zákazník, adresy, kontaktní osoby, bankovní účet a smlouvu. Poté, co je zákazník založen v Salesforce, se odešle notifikace o založení nového zákazníka spolu s jeho navázanými entitami. V OMS se pouze potvrdí založení zákazníka. Poté se teprve vše odesílá na OIP a dále do Siebel a OIS.

Je potřeba řešit dvojí mastering zákazníka v OMS a Salesforce. Z uživatelského pohledu může nastat v rámci OMS změna na zákazníkovi. Existuje tedy mimo zprávy o samozasmluvnění (která obsahuje veškeré informace pro založení zákazníka a dalších entit) i zpráva na aktualizaci záznamu zákazník. Tato notifikace se posílá z OMS přes WSO2 platformu do Salesforce. Salesforce si zprávu vyčítá a aktualizuje záznamu zákazník ve své databázi. Dále probíhá totožná notifikace do OIP, aby se mohla informace o změně dostat i do Siebel a OIS. Změna na entitě zákazník v Siebel již není ve druhé fázi možná.

4.4.6.2 Adresa / Address – ADD/EDIT (druhá fáze)

Adresa je podobně jako Account napojená přes WSO2 přímo do OMS přes fronty Q1 a Q2. Adres je více typů. Pokud se jedná o adresy napojené na zákazníka, pak vše probíhá v rámci klasické komunikace popsané výše. Proces je totožný pro založení zákazníka ručně v Salesforce i pro samozasmluvnění v OMS. Právní a Fakturační adresa podléhá kontrole 4 očí. Dokud není kontrola 4 očí dokončena, pak Salesforce neodešle notifikaci o její aktualizaci nebo vytvoření na WSO2. I adresu, která je napojena na zákazníka, lze upravit jak v Salesforce, tak v OMS. Pokud je adresa upravena v OMS probíhá zpětná notifikace přes WSO2 do Salesforce a dále pak klasickou cestou přes OIP do OIS a Siebel. Změna adresy, která je napojena na zákazníka, není již v systému Siebel možná.

Pokud však je daná adresa napojena na provozovnu, pak se řeší pouze v Siebel, protože provozovny zůstávají i v rámci druhé fáze v Siebel. Nemělo by tedy cenu masterovat entitu adresy provozovny v Salesforce. V Salesforce je však nutné vidět veškerá data včetně adresy provozoven. K tomuto účelu slouží rekonsiliace, která postupně aktualizuje záznamy v Salesforce a díky tomu je vše dostupné z jednoho místa. Pokud však uživatel chce upravit adresu provozovny, musí tuto změnu vyvolat v systému Siebel.

4.4.6.3 Bankovní účet / Payment Detail – ADD/EDIT (druhá fáze)

Bankovní účet neboli Payment Detail je masterován v Salesforce i OMS. Jeho integrace probíhá totožně jako u ostatních navázaných entit na zákazníkovi, tedy přes WSO2 do OMS a dále do OIP, OIS a Siebel. Pokud je bankovní účet změněn v OMS, pak se změna propisuje přes WSO2 do Salesforce a přes OIP do OIS a Siebel. Změna bankovního účtu není již v systému Siebel možná.

Bankovní účet, pokud je založený v Salesforce, podléhá kontrole čtyřech očí. Je tedy potřeba tuto kontrolu dokončit a až následně se notifikuje do WSO2.

4.4.6.4 Kontakt / Contact – ADD/EDIT (druhá fáze)

Kontaktní osoba je další entita, která povoluje dvojí masterování v Salesforce i v OMS. Funguje velice obdobně jako například adresa. Opět zde musíme rozlišit napojení na zákazníka a napojení na provozovnu.

Pokud je kontakt napojen na provozovnu, ukládá si vše Siebel a odesílá tyto vazby do OMS a OIS. Zpráva XML však nepočítá s novou vazební tabulkou mezi provozovnou a kontaktem, proto po vzoru vztahu provozovny a INP⁴⁸ ukládá do XML kontaktu pouze vazby na provozovny. Podobně to OIP řeší s Individuálními nastaveními produktů na smlouvách.

Pokud se jedná o napojený kontakt na zákazníka, pak se vše integruje jako v předchozích případech.

V rámci přechodu z druhé do třetí fáze projektu se již entita Kontakt integračně nemění. Není tedy potřeba ji dále sledovat.

4.4.6.5 Kontrakt / Contract – ADD/EDIT (druhá fáze)

Jak už bylo zmíněno výše, entita kontrakt je při samozasmluvnění posílána v rámci jedné velké zprávy obsahující všechny entity jako jsou zákazník, adresy, kontaktní osoby, bankovní účty a kontrakt.

Při vytvoření v Salesforce se musí daná entita propsat do systému Siebel tak, aby bylo možné v Siebel pokračovat se zadáním INP. Cesta do Siebel při manuálním zasmluvnění nebo aktualizaci záznamu je opět zvolena přes WSO2, OMS a OIP. Entita Kontrakt v sobě obsahuje atribut fáze smlouvy. Tato fáze slouží k pozdržení emailové notifikace pro zákazníka, aby se přihlásil do portálu, který v sobě OMS obsahuje. Je to z důvodu toho, aby obchodník stihl zadat domluvené individuální nastavení smlouvy do Siebel a tím vznikl v OMS kompletní a správný přehled. Pokud by se email odeslal hned a zákazník se přihlásil dříve, než obchodník stihne do Siebel zadat INP, pak by při objednávce viděl, že má všechny produkty za standardní ceny. Po zadání INP do Siebel, jejich kontrole a propsání do OIP a dále do OIS a hlavně OMS, se musí obchodník vrátit do Salesforce a zaktivnit smlouvu. Pokud tak učiní, OMS odesílá email s registračními údaji a zákazník se může přihlásit.

⁴⁸ [Viz. Kapitola – Individuální nastavení produktů / Contract Line – ADD/EDIT \(první fáze\)](#)

4.4.6.6 Individuální nastavení produktů / Contract Line – ADD/EDIT (druhá fáze)

Jak už je zmíněno výše, masterování INP zůstává v systému Siebel. I když se INP jako Quote (Contract) Line zadávají do Salesforce, je to pouze pro účely Business Modelu a případně pro generování smluvní dokumentace. INP se v této fázi ještě nikam neintegrují a musí se dvojitě zadávat jak do Salesforce, tak do Siebel. Po manuálním zadání INP do Siebel se klasickou cestou z první fáze projektu propisují přes OIP do OMS a OIS. Aby byla informace kompletní i v Salesforce, využívají se rekonciliace. Rekonciliace INP jsou velice náročné, protože došlo k přeměně produktového katalogu. Tyto změny musí být reflektovány a rekonciliace je pokaždé musí uplatnit, aby se do Salesforce odeslala správná data, dle nového produktového katalogu. Pokud je potřeba editovat INP, pak se v Salesforce musí provést dodatek smlouvy. V Siebel stačí pouze změnit přímo data v rámci INP dané smlouvy. Změna se pak propíše do ostatních systémů.

INP obsahuje stejně jako v první fázi vazby na provozovny, aby bylo na partnerské straně jasné, zda se na provozovně mohou produkty uplatnit. Tato vazba je v nezměněné formě přidána do XML zprávy INP a odesílána do ostatních systémů (OMS, OIS). Tuto vazbu je však nutné zobrazit v Salesforce. Zde opět fungují rekonciliace. Využívají vazební tabulky v Siebel, která existuje mezi INP a provozovnou a přenáší tuto informaci do Salesforce.

4.4.6.7 Provozovna / Workshop – ADD/EDIT (druhá fáze)

Provozovna je entitou, která je nadále masterovaná v Siebel. Veškeré integrační cesty zůstávají jako v první fázi, pouze se přidávají rekonciliace, aby bylo opět možné všechna data vidět v Salesforce. V Salesforce není možné vytvořit entitu provozovna.

4.5 Třetí fáze

V rámci třetí fáze dochází k ukončení provozu CRM Siebel. Všechny entity, které byly dosud masterované v Siebel, se přesouvají do Salesforce. Aby mohlo dojít k tomuto přesunu a zároveň nedošlo k přepsání starých systémů (požadavek od Společnosti),

muselo dojít ke vzniku nové aplikace – Legacy Adapter, která překládá stávající entity ze Salesforce do starého světa (tedy do OIP a dále do OMS či OIS). Jmenovitě se jedná o entity INP, Provozovna, Adresa provozovny a Produkt Provozovny (vztahová tabulka mezi INP a Provozovnou). Aby bylo vše správně integrováno, do Legacy Adapteru, přibyla i entita Kontrakt, tedy smlouva. Bylo by příliš komplikované, zanechat integrační tok entity Kontrakt tak, jak byl ve druhé fázi. V rámci třetí fáze si představíme novou aplikaci Legacy Adapter, její roli v projektu a opět zhodnotíme z business hlediska jednoduchost a časovou náročnost procesů.

4.5.1 Legacy Adapter

Legacy adapter je aplikace napsaná pomocí Spring Boot⁴⁹ frameworku. Jako databáze pro jednotlivé datové operace byla zvolena PostgreSQL⁵⁰

Legacy adapter je z části ETL nástroj, který transformuje a případně generuje data. Jeho provedení muselo být velice specifické, aby se dosáhlo stavu, kdy se v Salesforce nebude muset upravovat produktový katalog směrem ke starému stavu. Prostřednictvím Legacy adapteru se nepřenášel pouze produktový katalog, ale i další entity. To bude blíže popsáno v rámci integrační logiky třetí fáze. Tvorba Legacy adapteru byla poměrně chaotická, protože procesy, které obstarával Siebel, nebyly zdokumentované a často se stávalo, že se přeprogramovaly části aplikace. I přes tyto nezdary se Legacy Adapter podařilo implementovat tak, aby urychlil business procesy a umožnil zánik starého CRM.

4.5.2 Business procesy – třetí fáze

V této kapitole si popíšeme opět základní business procesy, které byly ve fázi 1 či 2. Třetí fáze se zásadně liší od paralelního režimu – není omezena existencí druhého CRM, kde se masteruje část dat. Zároveň umí využít Legacy adapter k přeložení dat do starého světa, aby vše procesně správně fungovalo (faktury, objednávky, samozasmluvnění, ...). Aby uživatelský přístup byl co nejkomfortnější, byly

⁴⁹ Spring boot je na Java postavený framework, který umožňuje snadno a rychle vytvářet webové aplikace. Zdroj: <https://spring.io/projects/spring-boot>

⁵⁰ PostgreSQL je open source databáze, která se používá na podobné účely. Zdroj: <https://www.postgresql.org/>

vytvořeny nové přístupy v rámci aplikace Salesforce, aby uživatel mohl pohodlně (jako v Siebel) pracovat s INP, aniž by se narušila logika CPQ. Řeč je o Contract Line Setting (CLS). CLS bude představeno v této kapitole a bude vysvětlen jeho přínos.

4.5.2.1 Vznik zákazníka (třetí fáze)

Vznik zákazníka se v SF příliš neliší od druhé fáze. Stále rozdělujeme zákazníky na prospekty a jejich tvorba je primárně přes flow v Salesforce, které dotahuje informace z ARES. Kolem zákazníka vzniká pouze problém na integrační straně v OIP. Tomu se podrobněji věnuje kapitola věnovaná integracím. Business proces jako takový tedy zůstává stejný jako u druhé fáze projektu.

4.5.2.2 Vznik provozovny (třetí fáze)

Novinkou ve třetí fázi je vznik provozovny. Do této chvíle se vše kolem provozoven řešilo v Siebel a do Salesforce se dostávaly informace pouze zprostředkovaně v rámci rekonciliací. Nyní se přesouvá vše do Salesforce, který musí zprostředkovat tyto informace do dalších systémů jak nových, tak i starých. Vnik provozovny je možný v Salesforce, případně v nových portálech samozasmluvnění, který je pro tuto práci nedůležitý, protože vše projde Salesforce a jedná se pouze o rozšíření případu užití. Pro provozovnu je důležitá připojená adresa, zařazení provozovny a produkt smlouvy.

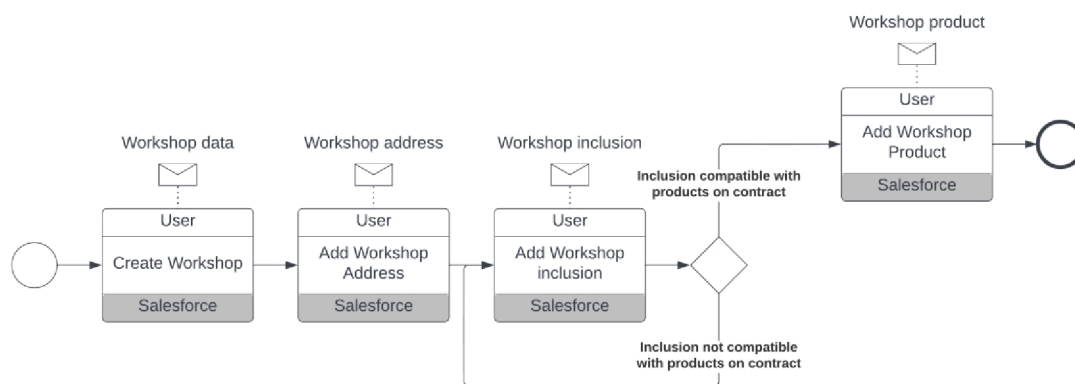
Provozovna vzniká pomocí flow, které zároveň nechá uživatele vyplnit adresu a zařazení. Po dokončení flow se vytvoří provozovna, adresa i zařazení provozovny. Všechny entity se automaticky posílají na integrace. Dalším krokem je přiřazení produktů na provozovnu. Pro tyto účely byla vytvořena Lightning Web Component⁵¹, která pomáhá uživateli vybrat správné produkty. Usnadněním zadávání, a jistou validací dat je fakt, že ke každému zařazení provozovny bylo určeno, jaký produkt je vhodný. V momentu zadání zařazení provozovny se kontroluje:

- Zda existuje produkt na smlouvě partnera.

⁵¹ [Viz. Kapitola – Lightning Web Component](#)

- Zda je některý z existujících produktů ze smlouvy kompatibilní se zařazením provozovny (dané zařazení obsahuje referenci na ten samý produkt v produktovém katalogu).
- Zda je produkt aktivní v produktovém katalogu.

Tato validace pomáhá s čistotou dat i celkovým procesem kompletního vytvoření provozovny. Po vytvoření vazby mezi provozovnou a produktem na smlouvě (INP) je objekt vazby notifikován do integrací. Tímto je proces vytvoření provozovny kompletní.



Obrázek 20 Vytvoření Workshopu, adresy a vazby na Contract Line (třetí fáze) – vytvořeno autorem

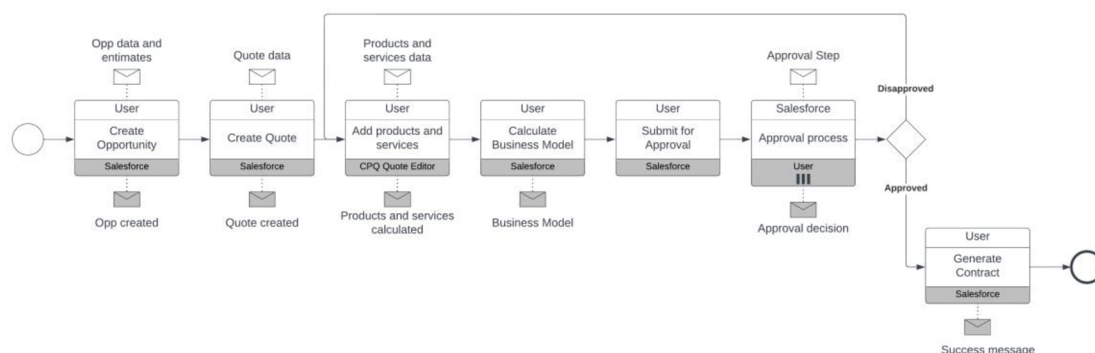
4.5.2.3 Vznik Smlouvy, INP a Business Modelu (třetí fáze)

Celkový proces zasmluvnění v Salesforce se průběhem příliš neliší od druhé fáze. V rámci business procesu se stále používá:

- Obchodní příležitost (Opportunita), která monitoruje celý průběh obchodu.
- Nabídka (Quote), která je nositelem produktů a služeb včetně jejich nastavení a cen, stejně tak jako nastavení celé smlouvy.
- Business Model, který přehledně zobrazuje profitabilitu.
- Schvalovací proces, který se opírá o hodnotu profitability či nastavení produktů.
- Proces zasmluvnění nabídky a vytvoření entity smlouvy (Contract).

V rámci posledního bodu, tedy procesu zasmluvnění nabídky, je změna oproti předchozí fázi. Změna je drobná a týká se vygenerování nové nabídky, která má sloužit pro rekonciliace. Tato „rekonciliační“ nabídka není již potřeba a nemusí se tedy generovat. Místo toho je stávající nabídka přiřazena jako primární ke smlouvě. Pokud se jedná o variantu, kde se dělá dodatek smlouvy, je původní nabídka přiřazena ke smlouvě jako historická a nová nabídka přiřazena jako primární. Tímto opět vzniká přehled revizí, které se na smlouvě staly. Není však již potřeba tvořit extra nabídku pro verzi smlouvy ze Siebel.

Další, a to velice podstatnou změnou je bod 7 z druhé fáze, tedy – dokončení zasmluvňovacího procesu v Siebel⁵², který již není potřeba. To, co se zadá do Salesforce nově platí pro všechny systémy, a to včetně aktualizace dat. Celý business proces se tedy výrazně zkracuje a redukuje se i případná chybovost při dvojitým zadáváním dat (do Salesforce a do Siebel).



Obrázek 21 Vytvoření kontraktu (třetí fáze) – vytvořeno autorem

INP je entita, kvůli které primárně Legacy adapter vznikl. Pro tuto entitu byly provedeny významné transformace. Sestavení transformací bylo poměrně náročné, protože se zde prolínaly klientské a partnerské INP, kde některá pravidla platila pro obě strany a některá zase výhradně pro jednu. Také se muselo uvažovat pořadí transformací. Dalším poměrně zásadním problémem při vzniku transformační logiky na Legacy adapteru byl fakt, že dokumentace ke starým systémům neexistovala. Nikdo neposkytl ucelený návod, jak jednotlivé INP fungují a co přesně

⁵² [Viz. Kapitola – Dokončení zasmluvňovacího procesu v Siebel](#)

znamenaají v kontextu jednotlivých systémů. Díky tomu se stávalo, že uprostřed vývoje se Legacy adapter předělával, protože se přišlo na nové skutečnosti. Nebylo ani výjimkou, že se takovýto hotfixing – neboli rychlé změny, děly při testování aplikace, kde jednotliví testeři znalí produktového katalogu odhalili chyby, nebo se na ně přišlo až po vyzkoušení fakturačního procesu apod.

Některé transformace v rámci INP jsou:

- Nastavení výchozích hodnot.
- Vymazání (respektive neodesílání) některých nových INP, kterým staré systémy nerozumí.
- Vygenerování nových INP na základě existence atributů či ostatních INP na smlouvě. V rámci úpravy produktového katalogu se redukovaly některé nezbytné INP, ale pro staré systémy se musely „obnovovat“.
- Přesouvání hodnot atributů z podřazených INP (služeb) do nadřazeného INP (produkt).

Transformace musely být několikrát optimalizovány, aby nedocházelo k cyklením a časovému zpoždění.

4.5.2.4 Contract Line Settings (CLS) a Reconfiguration Quotes

V původním CRM Siebel se mohlo libovolně manipulovat se smlouvou a „neřízeně“ měnit jak ceny. Bylo také možné odebrat (ukončovat) či přidávat služby nebo produkty. Tento přístup není v dané implementaci Salesforce ani v CPQ podporován. Není to tak, že by nebylo možné v rámci CPQ upravit již existující nabídku/smlouvu, ale v momentě úprav se ztrácí historie a revize nedávají smysl. Tento důvod dal tedy vzniknout dvěma přístupům, aby byly uspokojeny business požadavky.

Contract Line Settings neboli nastavení produktů smlouvy je entita, která upravuje na dobu určitou položky smlouvy – produkty či služby. Tím umožňuje uživateli zadat například dočasnou slevu kvůli opravě faktury nebo jakkoliv měnit ceny existujících produktů. Jedná se o objekt, který pomocí flow načte stávající produkty ze smlouvy. Uživatel si zvolí produkt a následně i službu, kterou chce měnit. V dalším kroku se pak mění cena pomocí slevy či přepsání hodnoty. Tento přístup umožňuje i další

konfigurace na existujících položkách smlouvy. Zároveň je vyžadováno časové ohraničení, aby uživatelé nepřepisovali produkty a služby na trvalo. K tomu by případně sloužil dodatek smlouvy, tedy nová revize. Pokud existuje alespoň jedno aktivní CLS, nelze vytvořit dodatek smlouvy.

Reconfiguration Quotes jsou nabídky, které obsahují odlišné nastavení smlouvy, produktů nebo služeb včetně jejich existence. Lze tedy například oproti původní smlouvě mít v rekonfigurační nabídce nějakou službu nebo celý produkt navíc. Reconfiguration Quote reaguje na požadavek, na který nestačí CLS. Jde především o dočasné odebrání či přidání služby nebo produktu na smlouvu. Aby pro uživatele byl daný proces co nejpříjemnější a bylo ho možné zopakovat, vytvoří se speciální verze nabídky, která umožňuje zásah do nastavení produktů a služeb. Tato nabídka se uloží a jediným přepnutím fáze na smlouvě se přepne primární nabídka na rekonfigurační a zpět.



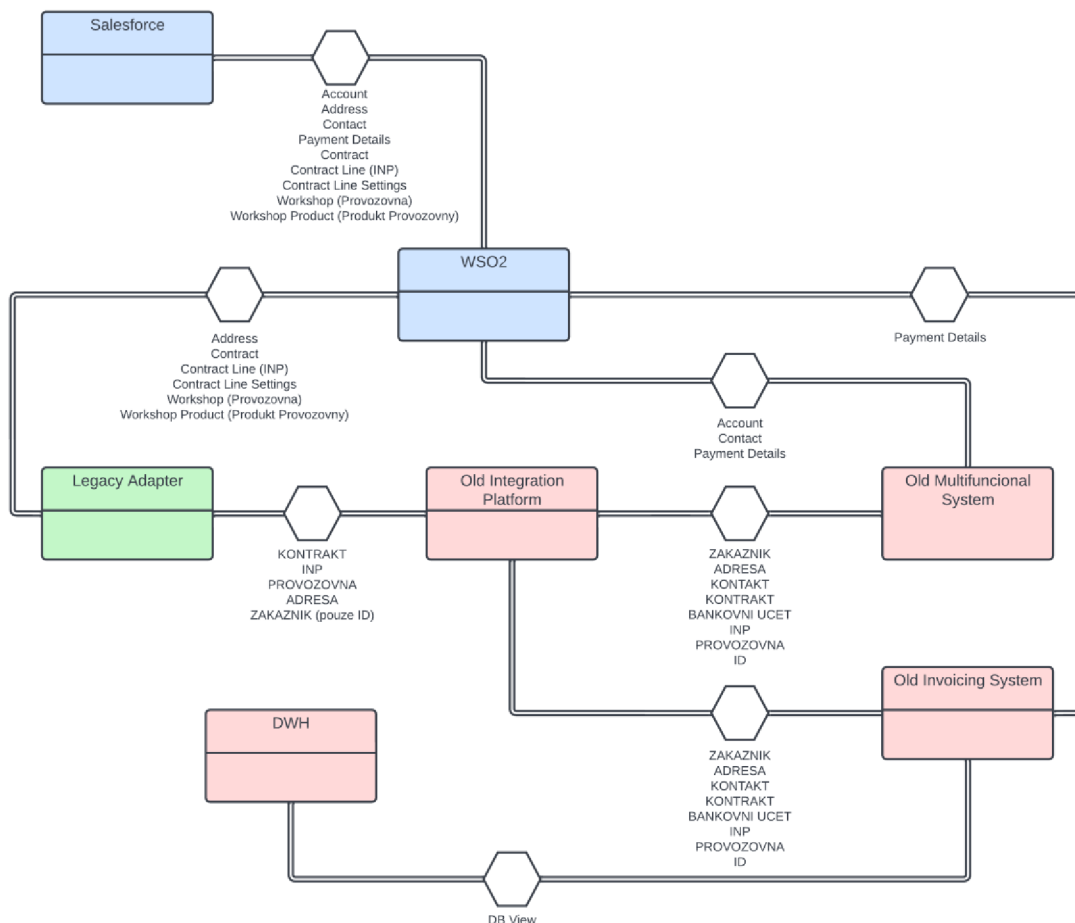
Obrázek 22 Vytvoření Rekonfigurační nabídky (třetí fáze) – vytvořeno autorem

4.5.3 Integrace systémů – třetí fáze

Finální podoba integrací je složitější než kdy dříve. Tento koncept je však potřebný, protože zde oddělujeme dva různé světy a oba si zachovávají své integrační platformy (WSO2 a OIP). Tato integrační logika přináší mnohé výhody, které se mohou uplatnit při nasazování dalších změn v integrační logice, nových portálů, aplikací, systémů apod. Je zde z větší části oddělen starý a nový svět. Pokud není potřeba napojit se napřímo do starého fakturačního systému a čerpat data přímo z jeho databáze, veškerý průpis dat zařídí Legacy adapter spolu s integračními platformami a OMS. V novém světě je to pak čistě WSO2.

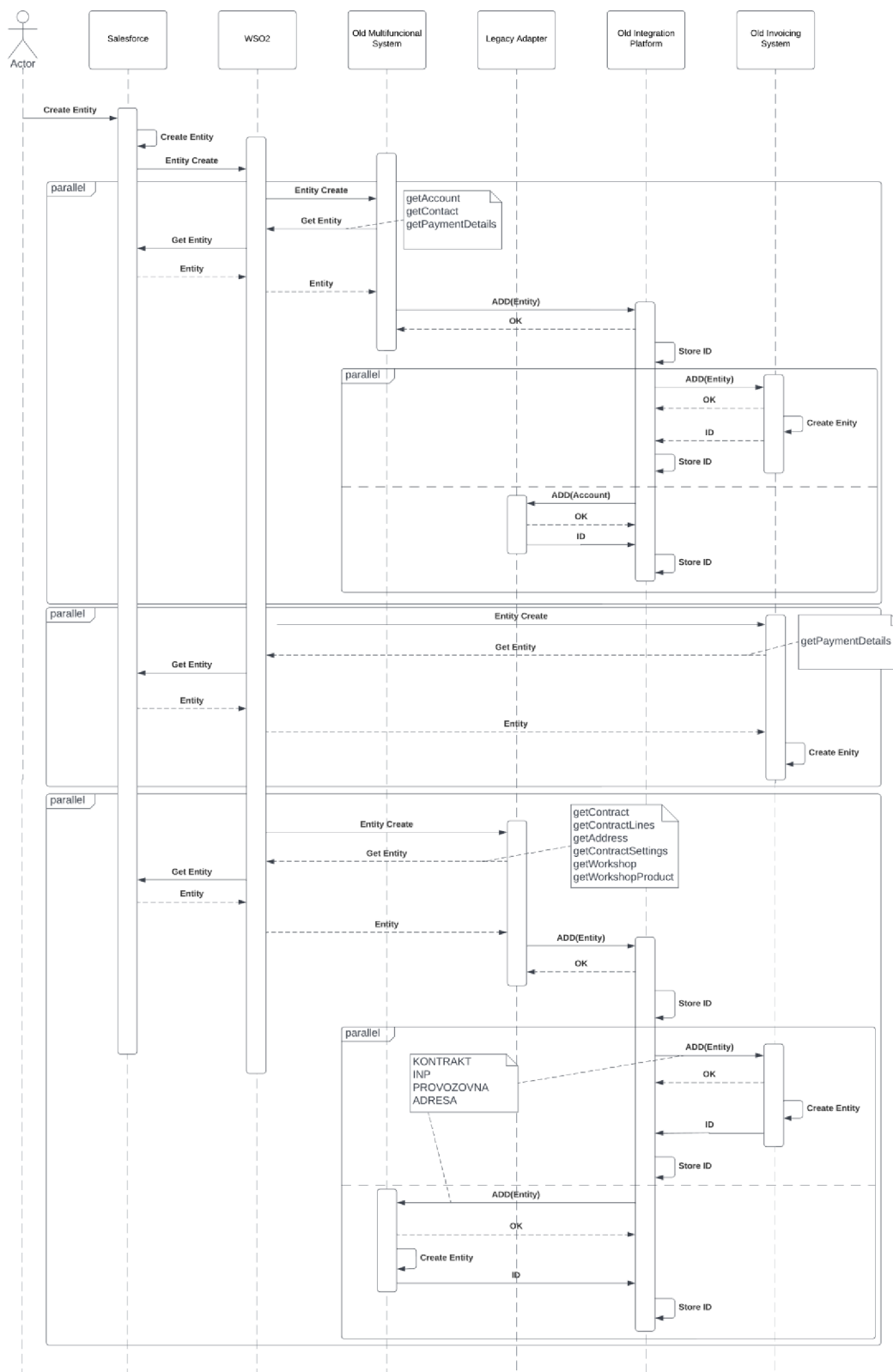
Jak už bylo řečeno OIP funguje na bázi asynchronního volání a překladu identifikátorů. Vzhledem k tomu, že se Legacy adapter napojoval přímo na OIP, je

yní novým konzumentem i odesílatelem integračních zpráv. Aby mohl v pořádku probíhat překlad identifikátorů, byl do tabulky IDs zaveden nový sloupec, do kterého byly naimportovány identifikátory všech entit, které komunikuje OIP s Legacy adapterem. Tímto je zaručen správný překlad identifikátorů na OIP.

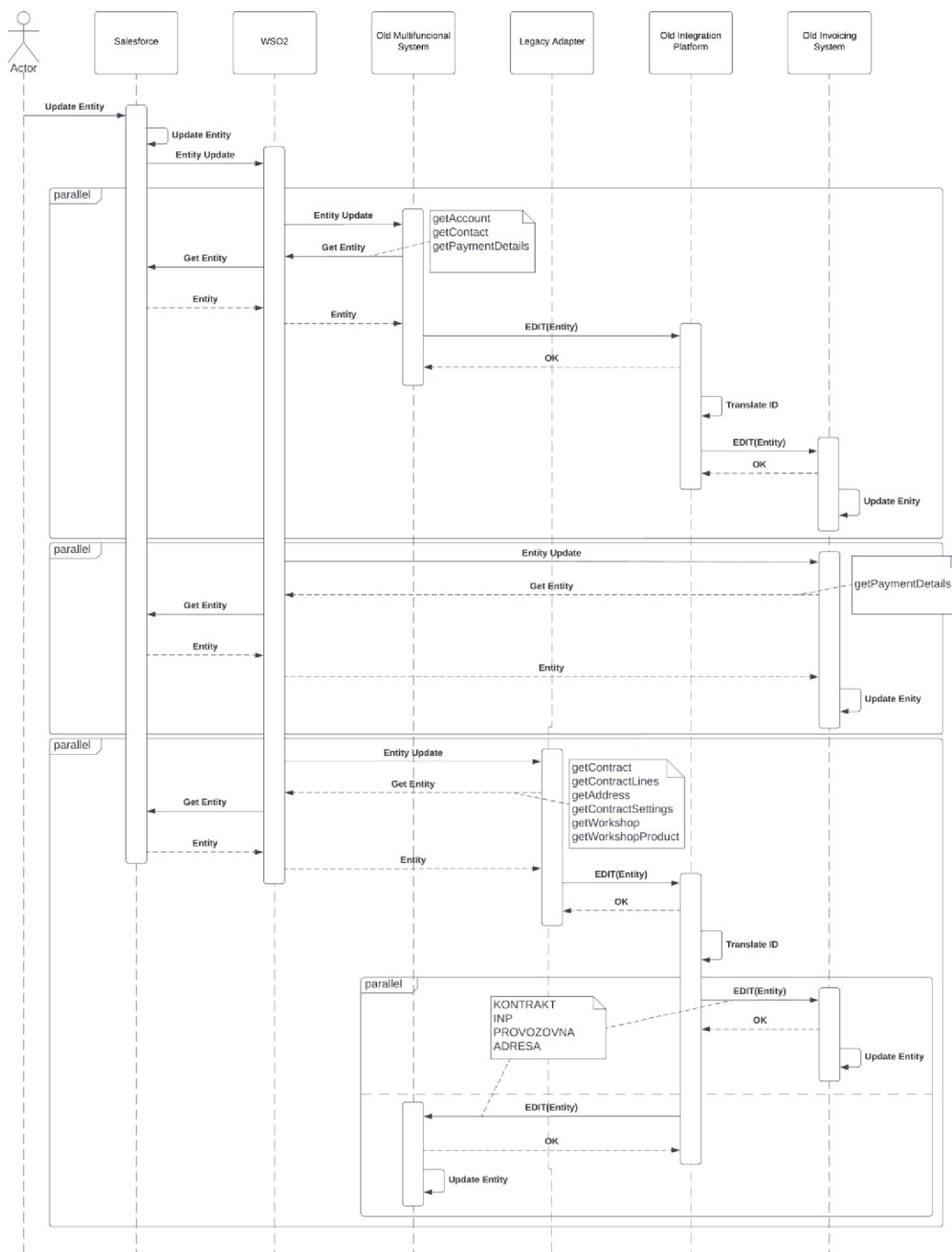


Obrázek 23 Conversation diagram provázanosti systémů ve třetí fázi - vytvořeno autorem

V této kapitole se znovu a již naposledy podíváme na jednotlivé entity, a jejich integrační toky.



Obrázek 24 Integrovaní toky entit při operaci ADD (třetí fáze) – vytvořeno autorem



Obrázek 25 Integrační toky entit při operaci EDIT (třetí fáze) – vytvořeno autorem

4.5.3.1 Zákazník / Account – ADD/EDIT (třetí fáze)

Entita zákazník je odesílána stále stejnou cestou, tedy ze Salesforce přes WSO2 do OMS, dále do OIP a starých systémů. Tento „starý“ přístup byl použit kvůli nedostatku času na projektu. Také daná integrace již v pořádku fungovala a nemělo

cenu ji předělávat. V rámci OIP však nastává změna. Jak už je řečeno v úvodu kapitoly Integrace systémů – třetí fáze, OIP musí komunikovat s Legacy adapterem a zároveň si ukládat jeho identifikátory, aby dokázal vše správně překládat do ostatních systémů. Problém nastává v momentě překladu identifikátoru nově založené entity zákazník. Stávající zákazníci mají v tabulce IDs v OIP nahrané svoje identifikátory. Nový zákazník se ale v Legacy adapteru nezakládá, a tedy nelze dát zpět vnitřní id zákazníka z Legacy adapteru.

Problém byl nakonec elegantně vyřešen. Máme totiž výhodu v tom, že při založení zákazníka přímo v Salesforce či jiném portálu Společnosti, se vždy dostane informace o novém zákazníkovi do Salesforce. Ten ji distribuuje dále. Nestane se tak, že například v rámci samozasmluvnění v OMS se zákazník dostane přímo do OIP. Nejdříve, jak je zmíněno v kapitole o samozasmluvnění, se vše odešle do SF. Tam se teprve zakládají všechny entity a až zpráva, obohacená o externí ID Salesforce, putuje do OMS. Díky tomu se zpráva do OIP rovněž dostává s identifikátorem Salesforce v atributu zprávy a ten ji odesílá do Legacy adapteru „k založení“. Legacy adapter si nepotřebuje držet všechny informace o entitě zákazník. Proto existuje pouze tabulka identifikátorů zákazníků, kterou dále Legacy adapter využívá. Právě do této tabulky zákazníků v Legacy adapteru se uloží externí ID Salesforce, které je obsaženo ve zprávě a zároveň se i vrátí v odpovědi do OIP, aby si mohl OIP do tabulky IDs uložit nové vnitřní ID Legacy adapteru. V podstatě se ale jedná o ID Salesforce. Protože OIP není navržen nejlépe, uvažuje se do budoucna o jeho vypnutí. Jeden z prvních kroků bude sjednocení tabulky IDs tak, aby všechny systémy používaly stejný externí identifikátor, který bude vytvářet Salesforce. Funkci uložení Salesforce externího ID si v rámci projektu převzaly všechny staré systémy. Tento zdánlivě zbytečný krok má však veliký dopad na další integrační zprávy. OIP tak pomalu bude ztrácet význam jako překladač identifikátorů, protože pro všechny nové entity zákazník bude v tabulce IDs stejná hodnota pro všechny systémy.

4.5.3.2 Adresa / Address – ADD/EDIT (třetí fáze)

Adresu už nemusíme z integračního hlediska rozdělovat na adresu přiřazenou k zákazníkovi a adresu přiřazenou k provozovně. Veškeré adresy jsou do starých

systemů nově odesílány přes Legacy Adapter. Kontrola čtyřech očí probíhá stále v Salesforce. Pokud není kontrola čtyřech očí v Salesforce dokončena, adresa se neodešle na integrace. V původním projektovém plánu Adresy nebyly integrovány přes Legacy adapter, ale měly být do daných systémů integrovány přímo přes WSO2 platformu. Bohužel, kvůli potřebným transformacím a problémům s neexistující CRM cross ID v návazných systémech se entita Adresa posílá přes Legacy adapter.

4.5.3.3 Bankovní účet / Payment Detail – ADD/EDIT (třetí fáze)

Bankovní účet byl zvolen jako jediná entita, pro kterou bylo možno použít novou integrační cestu – ze Salesforce přes WSO2 a přímo do OIS. Protože do OMS už synchronizována byla, řešilo se pouze, jak dostat bankovní účet do OIS. Dávalo by větší smysl odesílat ho jako ostatní entity do cílových systémů přes Legacy adapter. Jelikož není třeba transformace dat a Legacy adapter by tak jen přeposílal zprávu do OIP a ten dále do OMS a OIS, bude se bankovní účet přesouvat do OMS a OIP přímo, a to přes WSO2. Z pohledu pracnosti vše vyházelo lépe. Kdybychom zavedli entitu do Legacy adapteru, museli bychom:

- Naučit Legacy adapter novou entitu, pro kterou by musel řešit minimálně 2 tabulky – tabulku aktuálních bankovních účtů a tabulku fronty této entity.
- Do Legacy adapteru by se musely migrovat všechny stávající bankovní účty.
- OMS by se musel opět zrušit vyčítání fronty z WSO2 této entity a nově obnovit vyčítání z OIP.

Místo toho se vytvořil nový přístup do OIS a entita se začala posílat napřímo tam. Tento přístup, byť složitější, se ukázal jako správný, protože za dobu asi půlročního provozu s ním nebyl nikdy problém.

4.5.3.4 Kontrakt / Contract – ADD/EDIT (třetí fáze)

Entita Kontrakt je hlavní entita, která se posílá skrz Legacy adapter navazují se na ní dále entity Contract Line a Contract Line Settings. Jako taková prochází transformacemi:

- Mapování atributů, kde se zároveň z příchozího JSON zprávy vše převádí na XML, což je výsledná podoba pro OIP.
- Výpočet atributů, které OIP (respektive ostatní systémy) vyžadují. Tyto výpočty se týkají pouze formátu atributů, tedy doplnění například typu kontraktu, což lze vyčíst z jeho čísla.
- Nastavení výchozích hodnot, které jsou zakódované přímo v Legacy adapter, jedná se o atributy, které se již nevyužívají, ale na integracích starých systémů jsou stále mandatorní.

Jelikož v každém systému je každá jedna entita v rámci hierarchické struktury, je nutné myslet na pořadí. V Salesforce se v rámci integrací určuje pořadí snadno – entity vznikají postupně za sebou a není problém, že by do jiných systémů došla adresa, která je navázaná na zákazníka, který se ještě neodeslal. Tento princip není standardní, ale byl implementován v Salesforce za použití custom metadata types. Více k tomuto tématu je v kapitole Integrace v rámci Salesforce v teoretické části⁵³. Větší problém však nastává, pokud entity, které jsou na sobě závislé, vedou v rámci integrací jinými cestami. To nastává u práce entity Contract (Kontrakt), kde se očekává nejdříve založení entity Account (Zákazník) a až poté je na něj navázána smlouva. Jak už bylo zmíněno v předchozím textu, Legacy adapter se s touto skutečností musel vypořádat. Proto se velice hodí i skutečnost, že musí komunikovat s OIP kolem entity zákazník a to tak, že si ukládá do databáze Salesforce ID zákazníka. Díky tomuto kroku vzniká jistota, že se nový zákazník dostane do OIP, a tedy i do dalších systémů. Nutno podotknout, že integrační zpráva Contract už obsahuje ID zákazníka. V momentu, kdy se v Legacy adapteru ukládá nová smlouva je vždy zkontrolováno, zda daný nadřazený zákazník smlouvy je přítomen v tabulce zákazníků v Legacy adapter. Pokud ano, znamená to, že je již zpracovaný OIP a překlad ID i pořadí bude v pořádku. Pokud ne, pak si kontrakt a všechny jeho podřízené entity ukládá do fronty, kde „čekají“ až zákazník přijde v rámci integrací z OIP. Tento mechanismus je velice účinný a v momentu, kdy se reportuje tabulka,

⁵³ [Viz. Kapitola – Custom notifikace](#)

kde jsou uloženy všechny smlouvy, které se nepovedlo odeslat na integraci (fronta), se celá investigace případných problémů velice usnadňuje.

Tedy problém, který by jistě mnohdy nastával, protože integrace zákazníka „po staru“ je často pomalejší než nová integrace přes WSO2 a Legacy adapter, je vyřešen.

4.5.3.5 Individuální nastavení produktů / Contract Line – ADD/EDIT (třetí fáze)

INP, pro potřeby silných transformací proudí ze SF přes Legacy adapter a dále do OIP a starých systémů. Protože se jedná o hierarchickou strukturu, kde INP – Produkt je podřazen Kontraktu a INP – Služba je podřazená produktu, musí Legacy adapter dodržet pořadí zpráv tak, aby na OIP nedocházelo k chybám.

Bohužel nestačilo posílat entity ve správném pořadí. OIP a jeho vyčítání je nedokonalé a stávalo se, že i když Legacy adapter dodržel pořadí, OIP si vyčetl data v chybném pořadí a místo zpracování nahlásil chybu. Aby se tomuto chování mohlo předejít, byly vytvořena fronta pro entitu INP. Nejprve se uvažovalo časové zdržení mezi notifikací Kontraktu a INP v řádu několika málo sekund. OIP vše posílal již v pořádku, ale za to OIS nestíhal zpracovávat entity. Stávalo se, že OIS zpracoval některé entity dříve než jiné (aplikace má více vláken) a docházelo k předběhnutí při zápisu do databáze. OIS pak chyboval na neexistenci nadřazených entit. Aby se vše nakonec povedlo, musel si Legacy adapter stanovit „čekací lhůty“ na 1 min mezi Kontraktem a Produktem a 2 min mezi Produktem a Službou. V tomto režimu všechny integrace probíhají v pořádku.

4.5.3.6 Contract Line Settings – ADD/EDIT (třetí fáze)

Nová entita CLC – Contract Line Settings je řešena v rámci integrací tak, aby se dostaly všechny potřebné informace ze Salesforce do Legacy adapteru. Ten na jejich základě přeměnil potřebné atributy v rámci INP a odeslal je dále už ne jako CLC ale jako INP, do OIP. Jedná se o ukázkou toho, jak lze rozšiřovat stávající systémy, aniž by se muselo zasahovat do starých. Je to však samozřejmě omezené tím, co dokáže staré systému pojmout, aniž by si nemusely implementovat novou logiku. Tento objekt tedy vzniká v Salesforce a pomocí WSO2 se dostává do Legacy adapteru, kde se zpracovává. Zde jeho integrační cesta končí.

4.5.3.7 Provozovna / Workshop – ADD/EDIT (třetí fáze)

Provozovna se stejně jako smlouva odesílá do Legacy adapteru a dále do starých systémů. I když entita Provozovna byla potenciální adept na novou cestu do starých systémů, tedy napřímo přes WSO2 jako například Bankovní účet, nakonec se tak nestalo. Je to kvůli entitě Produkt provozovny, která vyžaduje transformaci a také kvůli poměrně složitému zpracování této entity ve starých systémech. Salesforce odesílá provozovnu poprvé od vypnutí starého CRM. Pro entitu Provozovna existují dvě podřízené entity – Adresa a Produkt provozovny. Entita provozovna je zase podřízená entitě Zákazník.

4.5.3.8 Produkt provozovny / Workshop Product – ADD/EDIT (třetí fáze)

Produkt provozovny je další a poslední entitou, která proudí ze Salesforce přes WSO2 do Legacy adapteru. Jedná se, podobně jako CLC, o entitu, která pouze upravuje INP, ale jako taková ve starém integračním světě neexistuje. Jak už bylo zmíněno, tato vztahová entita se integračně umísťuje na INP se kterou souvisí pomocí tagu, resp. seznamu navázaných provozoven. Legacy adapter toto zajišťuje a řeší veškeré s tím spojené problémy. Legacy adapter si ukládá veškeré produkty provozovny a řeší jejich aktivnost. Pokud přijde nějaký nový produkt provozovny či se zaktivní některý stávající, Legacy adapter přidá do tagu patřičná data na INP, kterého se Produkt provozovny týká a odesílá vše do OIP.

5 Shrnutí výsledků

Jako výsledek práce dodává autor formalizaci daného problému pomocí BPMN 2.0 diagramů, textu, nebo příkladu některých integračních zpráv. Díky formalizaci mohl být tento projekt implementován za poměrně krátkou dobu. I když bylo nutné řešení rozdělit do celkem tří fází, nakonec byla transformace CRM kompletní a výsledné řešení optimální. Optimalizace řešení spočívá především v tom, že se v časové tísni nemohly z různých důvodů implementovat změny ve starých systémech. Systémy, které zůstaly v dosavadní implementaci s prefixem „Old“, se budou dále odstraňovat a jejich funkcionalitu si postupně převezmou další systémy či samotný Salesforce. Dalšími výsledky práce jsou řešení všech problémů, které nastaly v průběhu celé implementace, navržené a použité optimalizace a urychlení stávajících procesů:

- Problémy v implementaci při takto velkém projektu jsou časté a mnohdy na jejich vyřešení není dostatek informací či času. Vynikajícím pomocníkem při řešení integračních problémů je vizualizace. Použití konverzačních a sekvenčních diagramů je dobrý způsob, který byl využit v průběhu projektu mnohokrát. Zároveň se tyto diagramy mohou použít jako dokumentace pro společnost.
- Optimalizace business procesů ve společnosti byla nedílnou součástí implementace nového CRM již od začátku. Dosavadní řešení bylo náchylné na chyby, neexistovaly validace, schvalovací procesy apod. Už s druhou fází projektu se podařilo chybovost snížit na minimum díky validaci vstupních dat, a to jak přímo v Salesforce, tak i z ostatních systémů. Zavedly se schvalovací procesy, které odbouraly nutnost přechodu do dalších aplikací jako je Sharepoint nebo řešení schvalování přes emaily. Některé optimalizace se bohužel dostavily až s kompletním vypnutím starého CRM, neb byly omezovány právě jeho existencí. Mluvíme tu především o optimalizacích v rámci Workshop Products a Contract Settings.
- Urychlení stávajících procesů přináší Salesforce oproti Siebel značné, už jen díky příjemnějšímu uživatelskému rozhraní, fulltextovému vyhledávání či přehledu informací na tzv. „klientské 360“. Největším a nejrobustnějším business procesem je zasmluvnění v Salesforce. Zdánlivě složitější než

původní procesy v Siebel. Je však nutno si uvědomit, že se jedná o několik původních business procesů dohromady včetně striktní validace a automatické kontroly produktového katalogu, schvalování nabídek, automatického generování čísla smlouvy apod. Celkový proces, který se nyní v Salesforce používá, dokázal zastoupit kontrolu 3 zaměstnanců Společnosti.

Poslední výsledek, který stojí za zmínku je autorem provedená case study, která se jako vzor může používat na projektech, jako byl tento. Nemusí nutně jít o obnovu CRM či výměnu jiného software. Case study jako taková byla průběžně měněna v rámci různých fází projektu. Jako taková může poskytovat návod, jak se odlišit od konkurence a získat tím výhodu na trhu.

6 Závěry a doporučení

Závěrem je nutno říci, že tato bezmála dvou roční implementace nového CRM byla dlouhodobým procesem zlepšování dosavadních funkcionalit a překonávání problémů. Každá fáze se dlouho připravovala. Mezitím se nejen stabilizoval systém po release předchozí fáze, ale vyvíjely se i různé optimalizace a další validace. Autor přináší již finální verzi CRM v kontextu business procesů. V rámci integrace se plánuje rozšiřovat portfolio navázaných systémů, pro které bude nové CRM hlavním zdrojem dat. Díky aplikaci Legacy Adapter je možno většinu omezení starého světa ignorovat. To je velkým benefitem celého navrženého řešení. Rozšiřující se požadavky businessu, které jsou nutné pro konkurenční schopnost na trhu, se mohou realizovat a tím dále posouvat možnosti Společnosti. I Legacy Adapter má svá logická omezení. Existují věci, které jsou nereprezentovatelné do starého světa a dokud nedojde k odstranění starých systémů, nebo k jejich rozšíření, nebude nikdy možno některé funkcionality implementovat a pomoci tak businessu.

6.1 Návrh na zlepšení dosavadního CRM

Návrhem na zlepšení Salesforce může být například podpora hierarchických smluvních podmínek. V praxi to vypadá tak, že mateřská firma uzavře se Společností smlouvu, která je většinou velice výhodná, protože firma garantuje značný odběr skrze své dceřiné společnosti. V implementovaném Salesforce nyní není možnost, jak na jednu smlouvu navázat více společností. Pokud navíc dojde ke změně ujednání smluvních podmínek (dočasná či trvalá sleva, akce na službu apod.), musí se nyní upravit znění u mateřské i všech dceřiných společností ručně. To se dá automatizovat kompletně v Salesforce. Není tedy třeba upravovat Legacy Adapter nebo stávající systémy.

7 Seznam použité literatury

- [1] What is CRM? *Salesforce.com* [online]. [vid. 2020-09-09]. Dostupné z: <https://www.salesforce.com/crm/what-is-crm/>
- [2] BÍLÝ, Marek. *Rozhrání pro integrace struktur B2B, B2C portálu a ERP* [online]. B.m., 2017 [vid. 2020-09-09]. Vysoká škola báňská - Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky. Dostupné z: <https://theses.cz/id/063ieh?info=1;isslret=C%3B;zpet=%2Fvyhledavani%2F%3Fsearch%3Dc%26start%3D33>
- [3] *Co je CRM systém? | Vyber CRM* [online]. [vid. 2020-09-09]. Dostupné z: <https://www.vyber-crm.cz/co-je-crm-system?fbclid=IwAR1-iVxuNbARekk0Hq7kLgmU0MC48aKzlr3SvS8W-JBIOGo97hCjMCcLYsc>
- [4] Marc Benioff. *Salesforce.com* [online]. [vid. 2021-10-17]. Dostupné z: <https://www.salesforce.com/company/leadership/bios/bio-benioff/>
- [5] BENIOFF, Marc a Carlye ADLER. *Behind the Cloud: The Untold Story of How Salesforce.com Went from Idea to Billion-Dollar Company-and Revolutionized an Industry*. B.m.: John Wiley & Sons, 2009. ISBN 978-0-470-53592-9.
- [6] *Co je cloud – definice | Microsoft Azure* [online]. [vid. 2021-10-17]. Dostupné z: <https://azure.microsoft.com/cs-cz/overview/what-is-the-cloud/>
- [7] What is API: Definition, Types, Specifications, Documentation. *AltexSoft* [online]. [vid. 2021-10-17]. Dostupné z: <https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/>
- [8] *What is AppExchange?* [online]. [vid. 2020-09-09]. Dostupné z: <https://appexchange.salesforce.com/mktcollections/curated/whatisappexchange>
- [9] Sales Performance Management Software. *Salesforce.com* [online]. [vid. 2022-01-08]. Dostupné z: <https://www.salesforce.com/eu/products/sales-cloud/overview/>
- [10] *Microsoft Outlook pro firmy – Microsoft* [online]. [vid. 2020-09-13]. Dostupné z: <https://www.microsoft.com/cs-cz/microsoft-365/outlook/outlook-for-business>
- [11] *Service Cloud Basics* [online]. [vid. 2022-01-08]. Dostupné z: <https://trailhead.salesforce.com/en/content/learn/modules/service-cloud-platform-quick-look>
- [12] SLACK. Slack - Features. *Slack* [online]. [vid. 2022-01-16]. Dostupné z: <https://slack.com/features>

- [13] Tableau - Our Products. *Tableau* [online]. [vid. 2022-01-16]. Dostupné z: <https://www.tableau.com/products>
- [14] *Customer 360 - Complete Salesforce Products & Software Suite - Salesforce EMEA* [online]. [vid. 2022-01-16]. Dostupné z: <https://www.salesforce.com/eu/products/>
- [15] Salesforce buys quote-to-cash company SteelBrick for \$360 million. *VentureBeat* [online]. 23. prosinec 2015 [vid. 2022-01-08]. Dostupné z: <https://venturebeat.com/2015/12/23/salesforce-buys-quote-to-cash-company-steelbrick-for-360m/>
- [16] *Salesforce Developers - Object List* [online]. [vid. 2021-10-17]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.object_reference.meta/object_reference/sforce_api_objects_list.htm
- [17] *Salesforce Help | Article - Person Account* [online]. [vid. 2021-10-17]. Dostupné z: https://help.salesforce.com/s/articleView?language=en_US&type=5&id=sf.account_person.htm
- [18] *Salesforce Developers - AccountContactRelation* [online]. [vid. 2021-10-17]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.object_reference.meta/object_reference/sforce_api_objects_accountcontactrelation.htm
- [19] *Opportunities* [online]. [vid. 2022-01-01]. Dostupné z: <https://help.salesforce.com/s/articleView?id=sf.opportunities.htm&type=5>
- [20] *Salesforce Help | Article - Sharing Settings* [online]. [vid. 2021-10-17]. Dostupné z: https://help.salesforce.com/s/articleView?language=en_US&type=5&id=sf.managing_the_sharing_model.htm
- [21] MICROSOFT 365 TEAM. *Klíčové ukazatele výkonnosti (KPI): Co jsou a jak se používají?* [online]. 10. září 2019 [vid. 2021-10-17]. Dostupné z: <https://www.microsoft.com/cs-cz/microsoft-365/business-insights-ideas/resources/what-are-kpis-and-how-to-use-them>
- [22] *Salesforce Developers - Compound Fields* [online]. [vid. 2021-10-03]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.object_reference.meta/object_reference/compound_fields_limitations.htm
- [23] *Salesforce Developers - Data Loader* [online]. [vid. 2021-10-03]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.dataLoader.meta/dataLoader/data_loader.htm
- [24] *Salesforce Developers - API calls, SOQL* [online]. [vid. 2021-10-17]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_soql.htm

- [25] *Salesforce Developers - Relationship among objects* [online]. [vid. 2021-10-17]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.object_reference.meta/object_reference/relationships_among_objects.htm
- [26] *Databázový software a aplikace | Microsoft Access* [online]. [vid. 2021-10-17]. Dostupné z: <https://www.microsoft.com/cs-cz/microsoft-365/access>
- [27] *Salesforce Developers - Field types* [online]. [vid. 2021-10-17]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.object_reference.meta/object_reference/field_types.htm
- [28] POLÁK, Robert. Posouzení výhod a nevýhod OOP databází vzhledem k relačním databázím [online]. 2009 [vid. 2022-01-09]. Dostupné z: <https://dspace.cuni.cz/handle/20.500.11956/20898>
- [29] *What is an API?* [online]. [vid. 2022-01-09]. Dostupné z: <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>
- [30] *What is Apex? | Apex Developer Guide | Salesforce Developers* [online]. [vid. 2022-01-09]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_intro_what_is_apex.htm
- [31] *Introduction to SOQL and SOSL | SOQL and SOSL Reference | Salesforce Developers* [online]. [vid. 2022-01-10]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.234.0.soql_sosl.meta/soql_sosl/sforce_api_calls_soql_sosl_intro.htm
- [32] *Salesforce Object Query Language (SOQL) | SOQL and SOSL Reference | Salesforce Developers* [online]. [vid. 2022-01-10]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.234.0.soql_sosl.meta/soql_sosl/sforce_api_calls_soql.htm
- [33] *Salesforce Object Search Language (SOSL) | SOQL and SOSL Reference | Salesforce Developers* [online]. [vid. 2022-01-10]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.234.0.soql_sosl.meta/soql_sosl/sforce_api_calls_sosl.htm
- [34] *Exception Class and Built-In Exceptions | Apex Reference Guide | Salesforce Developers* [online]. [vid. 2022-01-09]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.apexref.meta/apexref/apex_classes_exception_methods.htm
- [35] *Triggers | Apex Developer Guide | Salesforce Developers* [online]. [vid. 2022-01-10]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_triggers.htm

- [36] What is Unit Testing? Definition of Unit Testing, Unit Testing Meaning. *The Economic Times* [online]. [vid. 2022-01-09]. Dostupné z: <https://economictimes.indiatimes.com/definition/unit-testing>
- [37] *IBM - AND and OR logical operators* [online]. 8. března 2021 [vid. 2022-01-16]. Dostupné z: <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/en/spss-statistics/24.0.0?topic=expressions-logical-operators>
- [38] *Process Builder* [online]. [vid. 2022-01-16]. Dostupné z: https://help.salesforce.com/s/articleView?id=sf.process_overview.htm&type=5
- [39] *Flows* [online]. [vid. 2022-01-20]. Dostupné z: <https://help.salesforce.com/s/articleView?id=sf.flow.htm&type=5>
- [40] *Workflow* [online]. [vid. 2022-01-20]. Dostupné z: https://help.salesforce.com/s/articleView?id=sf.customize_wf.htm&type=5
- [41] *Execution Governors and Limits | Apex Developer Guide | Salesforce Developers* [online]. [vid. 2022-01-20]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_gov_limits.htm
- [42] CHURCHVILLE, Fred. What is user interface (UI)? Definition from SearchAppArchitecture. *SearchAppArchitecture* [online]. [vid. 2022-01-22]. Dostupné z: <https://searchapparchitecture.techtarget.com/definition/user-interface-UI>
- [43] FITZGIBBONS, Laura. What are front end and back end? Definition from WhatIs.com. *WhatIs.com* [online]. [vid. 2022-01-22]. Dostupné z: <https://whatis.techtarget.com/definition/front-end>
- [44] What is framework? - Definition from WhatIs.com. *WhatIs.com* [online]. [vid. 2022-07-10]. Dostupné z: <https://www.techtarget.com/whatis/definition/framework>
- [45] *What is Visualforce? | Visualforce Developer Guide | Salesforce Developers* [online]. [vid. 2022-07-10]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.pages.meta/pages/pages_intro_what_is_it.htm
- [46] *Introducing Lightning Web Components - Salesforce Lightning Component Library* [online]. [vid. 2022-07-10]. Dostupné z: <https://developer.salesforce.com/docs/component-library/documentation/en/lwc>

- [47] What Admins Need to Know About Salesforce Releases. *Salesforce Admins* [online]. 15. duben 2021 [vid. 2022-07-10]. Dostupné z: <https://admin.salesforce.com/blog/2021/salesforce-release-overview>
- [48] *What is ETL?* [online]. [vid. 2022-07-11]. Dostupné z: <https://www.oracle.com/cz/integration/what-is-etl/>
- [49] RENWICK, Priscila. Ultimate Introduction to Salesforce Integration. *Salesforce Ben* [online]. 2. únor 2022 [vid. 2022-08-14]. Dostupné z: <https://www.salesforceben.com/salesforce-integration/>
- [50] *Oracle Siebel CRM features and capabilities* [online]. [vid. 2022-04-03]. Dostupné z: <https://www.oracle.com/cz/cx/siebel/>
- [51] ADMIN. Cloud vs On-Premise: Stručná příručka. *Globema CZ* [online]. 13. prosinec 2017 [vid. 2022-04-03]. Dostupné z: <https://www.globema.cz/cloud-vs-premise-strucna-prirucka/>
- [52] *What is a Data Warehouse?* [online]. [vid. 2022-04-03]. Dostupné z: <https://www.oracle.com/cz/database/what-is-a-data-warehouse/>

Podklad pro zadání DIPLOMOVÉ práce studenta

Jméno a příjmení: Bc. František Konečný
Osobní číslo: I1700699
Adresa: Nebeská 414, Doubravčice, 28201 Český Brod, Česká republika
Téma práce: Návrh a technologická analýza prostřednictvím Sales Cloud
Téma práce anglicky: Design and technological analysis through Sales Cloud
Jazyk práce: Čeština
Vedoucí práce: doc. Ing. Hana Tomášková, Ph.D.
Katedra informačních technologií

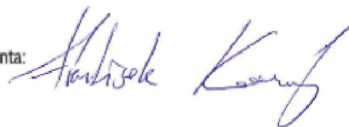
Zásady pro vypracování:

1. Úvod
2. Teoretická část
 - 2.1. Procesní analýza
 - 2.2. Technologická analýza
 - 2.3. Metody a technologie
 - 2.4. Sales Cloud
3. Praktická část
 - 3.1. Procesní analýza
 - 3.2. Případová Case study
 - 3.3. Výsledky technologické analýzy
 - 3.4. Popis využití Agilních přístupů v Case Study
4. Závěr
5. Diskuze
6. Reference

Seznam doporučené literatury:

1. Neustadt, I., & Arlow, J. (2016). *UML 2 a unifikovaný proces vývoje aplikací*. Computer Press, Albatros Media as
2. White, S. A., & Bock, C. (2011). *BPMN 2.0 Handbook Second Edition: Methods, Concepts, Case Studies and Standards in Business Process Management Notation*. Future Strategies Inc..
3. Ries, M., & Summers, D. (2016). *Agile Project Management: A Complete Beginner's Guide to Agile Project Management*. CreateSpace Independent Publishing.

Podpis studenta:



Datum: 3.5.2023

Podpis vedoucího práce:



Datum: 3.5.2023