Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Student	:	**Radek Theier**
Vedoucí bakalářské práce	:	**Ing. Lukáš Burkoň**
Oponent bakalářské práce	:	**doc. Jiří Feuerlicht, Ph.D.**

TÉMA BAKALÁŘSKÉ PRÁCE

Cloud computing a jeho model Platform-as-a-Service

ROK : 2010

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze kterých jsem čerpal.

V Praze dne ....

podpis

Poděkování

Rád bych na tomto místě poděkoval panu Ing. Lukáši Burkoňovi za pomoc s výběrem tématu a cenné připomínky v průběhu celého období psaní této práce.

Abstrakt

Bakalářská práce se zaměřuje na představení způsobu poskytování IT služeb pomocí sítě internet, pro který se v současnosti používá název cloud computing. Hlavní pozornost je věnována konkrétně modelu Platform-as-a-Service (PaaS), v jehož rámci jsou nabízena různá prostředí pro vývoj a následný provoz webových aplikací.

Cílem je představení tohoto relativně nového přístupu k vývoji a shrnutí výhod i možných překážek, které přináší v porovnání s klasickým vývojem aplikací. Součástí práce je i analýza současného trhu s PaaS platformami, která rozděluje dostupná řešení do tří základních kategorií. Účelem analýzy je usnadnit případným zájemcům o PaaS orientaci na trhu a pomoci jim vybrat vhodnou platformu tak, aby v maximální možné míře vyhovovala jejich konkrétním požadavkům.

V praktické části je na příkladu jedné z platforem demonstrován postup vytvoření jednoduché databázové aplikace na evidenci objednávek. V samotném závěru práce se autor pokouší odhadnout nejbližší budoucí vývoj celé oblasti cloud computingu i jeho modelu Platform-as-a-Service.

Abstract

This paper deals with the way of providing IT services via the Internet, which is nowadays known as cloud computing. It further focuses on the model Platform-as-a-Service (PaaS), in which vendors offer Web-based development-platforms for developing and deploying full programs online.

The main objective of the paper is to introduce this relatively new approach to developing applications as well as to shed light on related advantages and obstacles, thus contrasting with traditional ways of developing applications. The thesis also contains a market analysis of contemporary PaaS platforms, which divides possible solutions into three basic categories. The purpose of the analysis is to make it easier for potential customers to orientate on the market and help them to pick the right platform which suits their requirements best.

In the practical part there is a demonstration of the process of creating a simple database application using one of the platforms described earlier in this paper. At the end of the thesis the author makes an attempt to forecast the future of the whole area of cloud computing and the model Platform-as-a-Service.

1 Úvod 9

1.1 Vymezení tématu a cíle práce 9

1.2 Struktura práce 9

2 Cloud computing 10

2.1 Vysvětlení pojmu 10

2.2 Definice 11

2.3 Terminologie 12

2.4 Cesta ke cloudu 12

2.5 Virtualizace 15

2.6 Hlavní charakteristiky cloud computingu 15

2.7 Ekonomické aspekty 17

2.8 Green IT 19

2.9 Modely poskytování služeb 20

2.9.1 Software-as-a-Service 20

2.9.2 Platform-as-a-Service 22

2.9.3 Infrastructure-as-a-Service 22

2.10 Možné překážky 23

2.10.1 Bezpečnost 23

2.10.2 Lock-in 23

2.10.3 Legislativa 24

3 Platform-as-a-Service 24

3.1 Životní cyklus aplikace v rámci PaaS 25

3.1.1 Vývoj aplikací 25

3.1.2 Spolupráce 26

3.2 Architektura platformy 26

3.2.1 Multitenant aplikace 27

3.2.2 Architektura řízená metadaty 27

3.2.3 Definice dat a ukládání 28

3.2.4 APEX 29

3.3 Ekonomické přínosy vývoje aplikací pomocí PaaS 30

3.3.1 Nasazení a provoz aplikací 31

3.4 Možné obavy 32

4 Přehled dostupných platforem v rámci PaaS 33

4.1 Platformy pro vývoj aplikací v internetovém prohlížeči 34

4.1.1 ZOHO Creator (creator.zoho.com) 34

4.1.2 Qrimp (www.qrimp.com) 35

4.1.3 WaveMaker (www.wavemaker.com) 35

4.1.4 Wolf Frameworks (www.wolfframeworks.com) 36

4.1.5 Longjump (www.longjump.com) 36

4.1.6 OrangeScape (www.orangescape.com) 37

4.1.7 WorkXpress (www.workxpress.com) 37

4.1.8 Bungee Connect (www.bungeeconnect.com) 38

4.1.9 Force.com (www.force.com) 38

4.2 Platformy pro vývoj aplikací lokálně 40

4.2.1 Microsoft Azure (www.microsoft.com/windowsazure) 40

4.2.2 Google App Engine (code.google.com) 41

4.2.3 Stax Networks (www.stax.net) 42

4.2.4 Heroku (www.heroku.com), Engine Yard (www.engineyard.com) 42

4.3 Platformy pro tvorbu webových stránek a portálů 42

4.3.1 Sitemasher (www.sitemasher.com) 43

4.3.2 Cloudia CMS (www.cloudia.cz) 43

4.3.3 Homestead (www.homestead.com) 44

5 Vytvoření jednoduché aplikace 44

5.1 Vytvoření nové aplikace 45

5.2 Vytvoření formuláře 45

5.3 Naplnění tabulky daty 46

5.4 Propojení formulářů 47

5.5 Nastavení upozornění e-mailem 48

5.6 Formulářové akce 48

5.6.1 Akce při načtení (On Load) 49

5.6.2 Akce při potvrzení (Validate) 49

5.6.3 Akce při vložení hodnoty uživatelem (On User Input) 50

5.7 Pohledy (Views) 51

5.8 Publikování aplikace 52

5.9 Závěr 52

6 Možný budoucí vývoj 53

7 Závěr 55

8 Seznam použitých zdrojů 56

9 Terminologický slovník 61

10 Přílohy 63

10.1 Příloha A – Tabulka poskytovatelů z kapitoly 4.1 63

10.2 Příloha B – Tabulka poskytovatelů z kapitoly 4.2 64

10.3 Příloha C – Seznam obrázků použitých v textu 65

1 Úvod

1.1 Vymezení tématu a cíle práce

Oblast informačních a komunikačních technologií prošla během několika posledních desetiletí bouřlivým vývojem, který ani v současnosti nikterak neztrácí na síle. Jedním z posledních trendů na tomto poli je poskytování softwarových i hardwarových prostředků jako služeb pomocí sítě internet. Pro souhrnné označení této problematiky se dnes po celém světě používá anglický pojem *cloud computing*, který by se dal do češtiny přeložit pravděpodobně jako „výpočetní mrak“. Za tímto trochu tajemným názvem se skrývá řada modelů, vizí, nových přístupů apod., které mají potenciál změnit fungování IT oddělení v podnicích i státních organizacích tak, jak jej známe dnes. To si již uvědomila většina softwarových společností včetně těch největších a můžeme tak pozorovat velmi dynamický rozvoj trhu s produkty spadajícími do kategorie cloud computingu.

Tato práce si klade za cíl popsat základní pojmy a principy, které se váží k celé oblasti cloud computingu obecně. Hlavní část práce se zaměřuje konkrétně na jeden ze současných modelů cloud computingu, kterým je Platform-as-a-Service, zkráceně PaaS. Cílem je, aby čtenář získal alespoň základní přehled o tomto poměrně novém přístupu k vývoji webových aplikací, který přináší různé možnosti vytváření i relativně komplexních řešení pro uživatele ne-programátory. Zároveň by práce měla poskytnout přehled dostupných platforem spadajících do této kategorie za účelem pomoci případným zájemcům o PaaS se zorientováním se na tomto trhu a usnadnit jim tak výběr vhodné platformy pro jejich konkrétní potřeby.

Jelikož se jedná o poměrně novou oblast na poli informačních technologií, jistou překážkou při vypracovávání práce může být nedostatek kvalitní tištěné literatury vztahující se k tématu. Práce se tedy bude muset ve větší míře spoléhat na odborné články a další online zdroje.

1.2 Struktura práce

Ve své první části se práce zaměřuje na vymezení pojmu cloud computing jako takového a zabývá se jeho hlavními charakteristikami. Uvádí přehled výhod i některých problémů, které s sebou tento způsob využívání výpočetních zdrojů a aplikací přináší. Dále směřuje svou

pozornost na různé aspekty, které mohou ovlivňovat rozhodování potencionálních uživatelů, zda využít některých služeb nabízených v rámci cloud computingu. Součástí první kapitoly je také rozdělení oblasti do tří základních vrstev a jejich charakteristika.

Druhá část se detailněji zaměřuje na jeden ze tří základních modelů cloud computingu, konkrétně na Platform-as-a-Service. Práce se pokouší shrnout všechny výhody, které tento druh vývoje a provozu aplikací přináší oproti klasickému vývoji software, přičemž jsou využívány výsledky dostupných průzkumů, které proběhly ve společnostech využívajících některé PaaS platformy. Pozornost je věnována i technickému pozadí, které umožňuje vytvářet aplikace na platformách komerčních poskytovatelů.

Čtvrtá kapitola má za cíl analyzovat současný stav trhu s produkty Platform-as-a-Service. Pozornost je věnována jak velkým celosvětově známým IT společnostem, které nabízí svá řešení v této oblasti, tak i menším začínajícím projektům, jejichž nabízené služby mohou mít také své cílové zákazníky. Práce si klade za cíl obsáhnout celé spektrum dostupných možností, ačkoli to vzhledem k velmi dynamicky se rozvíjejícímu trhu s těmito produkty může být obtížné.

V předposlední prakticky zaměřené části bakalářské práce je na platformě společnosti ZOHO demonstrován postup vytvoření jednoduché databázové aplikace přímo v internetovém prohlížeči. Účelem je ukázka postupu, jakým mohou být aplikace na platformách PaaS vytvářeny.

V samotném závěru autor nejprve a podává svůj osobní náhled na možný budoucí vývoj celé oblasti cloud computingu a poté shrnuje zjištěné poznatky týkající se tématu, ke kterým v průběhu vypracovávání této bakalářské práce dospěl.

2 Cloud computing

2.1 Vysvětlení pojmu

Cloud computing se podle renomované analytické společnosti Gartner v současnosti nachází na vrcholu tzv. Hype křivky, která hodnotí vyspělost IT technologií a jejich připravenost na

implementaci. [GAR2009] To znamená, že je předmětem velkého zájmu a očekávání, ale k všeobecnému osvojení a přijetí by podle analytiků mělo dojít až v horizontu 2-5 let. Podle stejné křivky však tomuto kroku předchází ještě období zklamání a deziluze, které tedy můžeme v nejbližší budoucnosti očekávat.

Ačkoli se tedy pojem cloud computing skloňuje ve všech pádech, uvést nějakou jednoduchou a dostatečně srozumitelnou definici není vůbec jednoduché. Diskuze se vede už jen nad tím, zda je cloud computing pouze buzzword či toto slovní spojení stojí za označením nějaké úplně nové technologie. Podle George Reese není buzzword o nic víc než pojem „the Web“. [REE2009, str.1] Nepřináší nic zásadně nového na poli technologií. Jedná se spíše o změnu přístupu k budování IT infrastruktury za pomoci řady technologií, které samostatně existují již několik let. Spoluzakladatel společnosti Oracle Larry Ellison dokonce tvrdí, že se jedná pouze o „módní blábol“ označující vše, co se již stejně na poli IT děje. [STA2007] Ještě dál zachází zakladatel projektu GNU Richard Stallman, který cloud computing nazývá stupiditou a pastí a varuje před používáním webových aplikací z důvodu ztráty kontroly.[STA2007]

2.2 Definice

Jednu možnou definici pojmu cloud computing nabízí již výše zmiňovaná společnost Gartner. Dle ní pojem označuje *„způsob práce s počítačem, který je škálovatelný, přizpůsobitelný možnostem IT a je externím zákazníkům dodáván jako služba pomocí internetových technologií“*.[PLU2009]

S jinou definicí přichází Národní institut standardů a technologie (NIST), což je laboratoř měřících standardů při ministerstvu obchodu USA:

„Cloud computing je model umožňující pohodlný, na vyžádání dostupný síťový přístup ke sdíleným výpočetním zdrojům (např. sítím, serverům, úložištím, aplikacím a službám), které mohou být rychle poskytovány a opět uvolňovány s minimálním úsilím nutným k řízení této operace nebo s minimální interakcí ze strany poskytovatele služby. Takový model distribuce zaručuje trvale dostupnost potřebných výpočetních zdrojů.“ [MEL2009, str.1]

Pro snazší pochopení bývá celá problematika velmi často vysvětlována na příkladu distribuce elektřiny či vody. [CAR2003, str. 7] Dříve měla každá továrna svůj vlastní zdroj elektrické energie. Postupem času se však přišlo na to, že elektřinu je možné přenášet i na delší vzdálenosti a že se vyplatí postavit velké elektrárny na vhodných místech (například u velkých řek). Pro továrny pak bylo mnohem výhodnější si elektřinu kupovat, než si ji vyrábět. Paralelně k tomuto příkladu jsou dnes výpočetní zdroje soustředěny do obrovských datových center, odkud si podniky na základě své aktuální potřeby tyto zdroje prostřednictvím internetu pronajímají.

2.3 Terminologie

Cloud computing zahrnuje jak aplikace poskytované jako služby prostřednictvím internetu, tak i hardware a software v data centrech, pomocí kterých jsou tyto služby poskytovány. Tento hardware a systémový software se souhrnně označuje jako *cloud*. Pokud je tento cloud veřejně přístupný pay-as-you-go formou, nazýváme ho *public cloud*. Jestliže jsou prostředky tohoto public cloudu nabízeny jako služba, mluvíme o *utility computingu*. Interní podniková datová centra, která nejsou veřejně přístupná, bývají označována jako *private clouds* a v modelu cloud computingu se neuvažují.[ARM2009, str.4] Dále ještě rozlišujeme tzv. *community cloud*, kdy je infrastruktura cloudu sdílena několika organizacemi a má za cíl podporovat jejich společné zájmy a cíle (např. bezpečnost, politiku,…). Může být spravován přímo jednotlivými organizacemi či jinou třetí stranou. Posledním typem cloudu je tzv. *hybrid cloud*, který je kompozicí dvou a více cloudů (public, private nebo community).[MEL2009, str.2]

2.4 Cesta ke cloudu

Způsob řešení informatiky v podnicích prošel během zhruba 50 let od začátku užívání počítačů až po současný stav řadou změn a proměn. Tento vývoj modelů dodávky ICT služeb lze zachytit spirálou, u které můžeme pozorovat vždy po určitých časových úsecích návrat k modelům dřívějším, avšak díky technologickému pokroku na mnohem vyšší kvalitativní úrovni. Každý z modelů hledá odpověď na otázku *„kdo a jakým způsobem má zajišťovat vývoj aplikací a kdo a jakým způsobem má zajišťovat provoz aplikací, resp. celého informačního systému podniku“*.[VOR2008, str. 44]

Počítače začaly být ve větší míře v podnicích využívány na konci 60. let 20. století. Protože však cena výpočetní techniky byla v této době značně vysoká a pro podniky nedostupná, vznikala speciální servisní střediska, která se specializovala na dávkové zpracování dat. Zákazník na nosiči přinesl data do střediska a ta byla za určitou dobu (hodiny, dny) zpracována. Tento model měl své výhody i nevýhody. Podnik nemusel vlastnit své vlastní výpočetní středisko a mohl se soustředit jen na svůj hlavní předmět podnikání. Oproti tomu specializované servisní středisko mohlo díky velkému počtu zákazníků užívat výhody úspor z rozsahu. Nicméně celý proces zpracování dat byl velmi zdlouhavý a nebyl interaktivní.

Postupem času začaly růst požadavky na rychlost zpracování dat a externí zpracování podnikům přestalo vyhovovat. Tehdejší technologie neumožňovaly přenášení dat na dálku, proto si podniky začaly budovat své vlastní ICT oddělení, což bylo možné i díky snižování cen výpočetní techniky. Z důvodu nedostupnosti vhodného typově aplikačního software byly podniky nuceny si tento software vytvářet vlastními silami za pomoci vlastních programátorů.

V 80. letech minulého století požadavky na aplikace neustále rostly a pro podniky se stalo vytváření těchto aplikací za pomocí vlastních zdrojů velmi nákladné a časově náročné. Ve velké míře se začal využívat různý typově aplikační software. Podnik si nakoupil svou vlastní infrastrukturu a software od externích dodavatelů, nechal si jej naimplementovat a pak už celý systém provozoval vlastními silami. Tento model bývá označován „software jako licence“.

Se splasknutím tzv. „internetové bubliny“ (dot-com-bubble) v roce 2001 byla většina podniků nucena zredukovat své náklady a náklady na ICT, které tvořily nemalou část celkových nákladů podniků, nebyly výjimkou. Podniky byly totiž opět nuceny investovat velké sumy peněz do neustále se vyvíjejících ICT technologií, které ale vůbec nemusely souviset s hlavním předmětem podnikání. Tyto výdaje byly navíc obtížně kontrolovatelné. Další nemalou položku v nákladech tvořily mzdy IT zaměstnanců, z nichž někteří byli navíc potřeba jen nárazově. To vše vedlo k hledání nového modelu přístupu k řešení dodávky ICT služeb.

Řada podniků se vydala cestou outsourcingu. Veškeré své zdroje a aktivity (hardware, software, IT pracovníci) související s informačním systémem začaly převádět na jinou společnost, od které si pak tyto zdroje nakupovaly jako službu. To přináší podniku podobné výhody, jako kdysi externí dávkové zpracování, např. podnik nemusí investovat do ICT infrastruktury a může se soustředit na svou hlavní podnikatelskou činnost. Další výhodou outsourcingu je možnost měnit

objem nakupovaných služeb dle potřeby podniku. [BRU1998] Outsourcing však ne vždy přinášel požadované efekty v podobě úspor za informatické služby na straně podniku a úspor z rozsahu na straně poskytovatele těchto služeb. Důvodem byl často outsourcing příliš individuálních aplikací, které poskytovatel mohl nabízet pouze jedné společnosti. Další potencionální problémy představovaly smlouvy mezi poskytovatelem a podnikem, které musely ošetřovat řadu oblastí (help-desk, objem služeb, funkcionalita atd.).[VOR2008]

S postupným rozšiřováním kapacit internetového připojení se začal rozšiřovat model „software jako služba“, kdy si podnik aplikace „pronajímá“ prostřednictvím internetu. Tento model vychází z konceptu ASP (Application Service Provider), dnes se používá označení SaaS (Software as a Service). Model SaaS je v současnosti řazen do cloud computingu a tvoří jednu z jeho základních vrstev. Jeho detailnější popis je uveden níže v této práci.

Masivnějšímu rozšiřování ideji SaaS a vůbec celého cloud computingu musela předcházet fáze budování obřích datových center, která by byla schopna bez problémů obsluhovat všechny požadavky stávajících i budoucích uživatelů. Tato centra jsou z důvodů obrovské energetické náročnosti a velkých požadavků na chlazení budována ve strategických lokalitách, například u velkých řek nebo v místech s chladnějším podnebím. Velká datová centra tedy vznikla například v USA ve státě Oregon poblíže řeky Columbia a třeba Microsoft se rozhodl vybudovat část své infrastruktury na Sibiři.[KOD2008] Společnost Google se ohledně svých datových center ze strategických důvodů chová velmi tajemně. Je však známo, že neustále buduje další a další centra po celém světě, v současnosti především v Asii.[MIL2008]

Ve vývoji modelů dodávky ICT služeb můžeme tedy pozorovat neustálý rozpor mezi centralizací a decentralizací výpočetních zdrojů. Model cloud comptutingu se přiklání opět na stranu centralizace. Může se to zdát nelogické, v době, kdy dnešní osobní PC jsou schopny dosahovat výkonů na úrovni dřívějších superpočítačů a kdy navíc jejich ceny neustále klesají, budovat obří datová centra a přesouvat do nich většinu našich výpočetních požadavků a dat. Cloud computing ovšem kombinuje výhody všech předchozích modelů. Umožňuje poskytovatelům infrastruktury či aplikací v cloudu využívat výhody úspor z rozsahu a uživatelům služeb cloudu naopak nechává prostor pro to, aby se věnovali hlavnímu předmětu podnikání. A díky existenci internetu je dostupnost zdrojů v cloudu rychlá a interaktivní.

2.5 Virtualizace

Cloud computing by nikdy neexistoval bez technologie zvané virtualizace. Ta přidává jakousi virtuální vrstvu (hypervisor) mezi hardware a software a umožňuje tak přistupovat k výpočetním zdrojům jinak, než fyzicky existují. Na jednom fyzickém počítači tak může například běžet několik virtuálních počítačů (strojů) s rozdílnými operačními systémy nebo jedna fyzická paměť může být rozdělena na několik virtuálních.

První pokusy s virtualizací se datují již do 60. let 20.století, kdy firma IBM testovala tuto technologii na svých tehdejších sálových počítačích typu *mainframe*. V současné době je lídrem ve tvorbě virtualizačního software americká firma VMware[1]. Alternativním produktem je například opensource projekt Xen[2] vyvinutý na univerzitě v Cambridgi.

Ačkoli existuje několik druhů virtualizace (např. hardwarová virtualizace, para-virtualizace či virtualizace na úrovni operačního systému), můžeme najít několik obecných výhod, které tato technologie přináší: [HAJ2007]

- umožňuje plné využití hardwarového výkonu a tím sloučit více služeb na méně serverů
- umožňuje provozovat více operačních systémů na jednom fyzickém serveru
- zjednodušuje zálohování, obnovu záloh, disaster recovery či migraci systému
- umožňuje dynamické přidělování výkonu (navyšování či snižování)

Poskytovatelům cloudu tak dává možnost obsloužit velké množství zákazníků a udržet své náklady v rozumných mezích díky plnému využití stávající infrastruktury.

2.6 Hlavní charakteristiky cloud computingu

Při popisu cloud computingu je nezbytné uvést klíčové charakteristiky, které vymezují jeho vztah k ostatním tradičním přístupům řešení informatiky a naopak rozdíly, které ho od nich odlišují. Pět takovýchto hlavních charakteristik uvádí Národní institut pro technologie (NIST): [MEL2009, str.1]

[1] www.vmware.com
[2] www.xen.org

On-demand self-service

Toto do češtiny obtížně přeložitelné spojení v podstatě popisuje způsob, jakým se uživatel může dostat k dodatečným výpočetním zdrojům. Důležité je, že se jedná o jednostrannou akci uživatele, u které není zapotřebí interakce s žádným zástupcem poskytovatele služeb.

Široký přístup k síti

Zdroje jsou k dispozici přes síť a přístupné pomocí standardních mechanismů, které podporují používání různorodých klientských platforem (např. mobilní telefony, laptopy, PDA).

Sdružování zdrojů (Resource pooling)

Výpočetní zdroje na straně poskytovatele jsou sdružovány za účelem obsluhování mnoha uživatelů, přičemž je využíván tzv. multi-tenant model. Rozdílné fyzické či virtuální zdroje jsou dynamicky přidělovány nebo přerozdělovány podle požadavků uživatele. Uživatel obecně nemá tušení, kde přesně se nachází výpočetní zdroje, které právě využívá. Na vyšší úrovni abstrakce však může být alespoň schopen určit, v jakém státě či v jakém data centru se nachází. Příkladem těchto zdrojů mohou být ukládání, zpracování, paměť nebo virtuální stroje.

Velká elasticita

Někdy se také používá označení „škálovatelnost“ (scalability). Výpočetní zdroje jsou poskytovány rychle a pružně v závislosti na aktuální potřebě. V době největšího náporu jsou přibrány další zdroje a naopak v době útlumu, když už nejsou potřeba, jsou bez jakýchkoli sankcí jednoduše odebrány. Toto škálování může probíhat zcela automaticky a nebo manuálně, nejčastěji pomocí webové aplikace. Důležitým aspektem je, že koncovému uživateli se poskytované výpočetní zdroje jeví jako nekonečné a mohou být „nakupovány“ kdykoli a v jakémkoli množství.

Měřená služba

Systém cloudu automaticky kontroluje a optimalizuje využívání zdrojů použitím měření na různých úrovních abstrakce v závislosti na typu služby (např. ukládání, zpracování, kapacita připojení). Využívání zdrojů může být monitorováno, kontrolováno či hlášeno a umožňuje tak oboustrannou transparentnost jak pro poskytovatele, tak pro zákazníka využívané služby.

Mimo těchto pět hlavních charakteristik existují ještě další, jako například **účtování pay-as-you-go formou**, kdy je využití služeb zpoplatněno například počtem cyklů CPU nebo množstvím ukládaných dat za určitou dobu, v závislosti na typu služby a na poskytovateli. Důležité ovšem je, že uživatel platí vždy jen to, co opravdu spotřebuje.

2.7 Ekonomické aspekty

Cloud computing nepřináší žádné velké technické novinky, pouze vhodně kombinuje již existující prostředky a to za účelem maximalizace užitku při co nejnižších nákladech. Někdy bývá také popisován jako „přeměňující kapitálové výdaje na provozní náklady“.[ARM2009, str. 10] Oproti klasickému řešení IT v organizacích totiž odpadá nutnost rozsáhlých investic do IT infrastruktury.

Při rozhodování vedení společnosti o způsobu řešení informatiky v podniku je potřeba vzít do úvahy mnoho kritérií od velikosti, počtu zaměstnanců, předmětu podnikání, stávajícího stavu IT až po vizi budoucího směřování společnosti. Jednoduché porovnání tří možných přístupů nabízí ve své knize George Reese [REE2009, str. 12-13]:

	Interní IT	**Outsourcing**	**Cloud**
Počáteční kapitálová investice	značná	průměrná	zanedbatelná
Průběžné náklady	průměrné	značné	závislé na užívání
Časová náročnost přidání dodatečných zdrojů	značná	průměrná	žádná
Flexibilita	omezená	průměrná	flexibilní
Požadavky na vlastní zaměstnance	značné	omezené	průměrné
Spolehlivost	různá	vysoká	průměrná až vysoká

Tab. 2.7: Srovnání různých řešení IT v organizacích

Jednotlivé kategorie použité v tabulce mají tento význam:

Počáteční kapitálová investice – kolik prostředků je nutné vynaložit na vybudování potřebné infrastruktury či provedení změn v ní
Průběžné náklady – náklady na správu infrastruktury včetně nákladů na lidské zdroje a dodatečné služby
Časová náročnost přidání dodatečných zdrojů – jak dlouho trvá přidat do stávající infrastruktury novou komponentu
Flexibilita – schopnost infrastruktury reagovat na neočekávané výkyvy v požadavcích na zdroje
Požadavky na vlastní zaměstnance – kolik a jak vzdělaných a zkušených pracovníků je potřeba k řízení infrastruktury
Spolehlivost – pravděpodobnost, že poskytovaná služba bude k dispozici 24/7

Hlavní výhoda řešení v rámci cloud computingu tkví ve velké flexibilitě (elasticitě), kdy nabízí možnost pružně reagovat na aktuální požadavky na infrastrukturu. Z ekonomického hlediska se tedy můžeme vyhnout situaci předimenzování nebo naopak poddimenzování výpočetních zdrojů. V případě předimenzování platíme zbytečně za zdroje, které nevyužíváme. Na druhé straně v případě poddimenzování nám hrozí ztráta uživatelů (zákazníků) z důvodu nedostupnosti naší služby v určitém čase. Někdy se ani při klasickém využívání interních IT zdrojů není možné předimenzování vyhnout, jak uvádí následující příklad:

„Předpokládejme, že jsme schopni odhadnout, že námi nabízená služba vyžaduje v poledne v době vrcholné poptávky 500 serverů, o půlnoci však již jenom 100. Průměrný požadovaný počet serverů za celý den je tedy 300 a tomu odpovídá 7 200 server-hodin (300 x 24). Pokud ale musíme naše výpočetní zdroje dimenzovat tak, aby zvládly i vrchol poptávky, musíme vlastnit 500 serverů a platit tedy 12 000 server-hodin (500 x 24), což je 1,7 krát více než v průměru potřebujeme.“ [ARM2009, str. 10]

Jedná se samozřejmě pouze o modelový příklad, je na něm však možné dobře demonstrovat hlavní výhodu cloud computingu – elasticitu. Ve skutečnosti musí každý uživatel pečlivě zvážit, zda je jeho aplikace vůbec schopna využívat potencionálních výhod cloud computingu.

Nikde není řečeno, že vzdání se vlastních serverů a namísto toho využívání infrastruktury v rámci cloud computingu nutně povede ke snížení nákladů na tyto zdroje. Hlavním bonusem cloudu je ovšem to, že v dnešním turbulentním ekonomickém prostředí, kdy je velmi obtížné plánovat v delším časovém horizontu, nemá podnik svázané ruce nějakou velkou investicí do IT infrastruktury a může pružně reagovat na aktuální dění na trhu.

2.8 Green IT

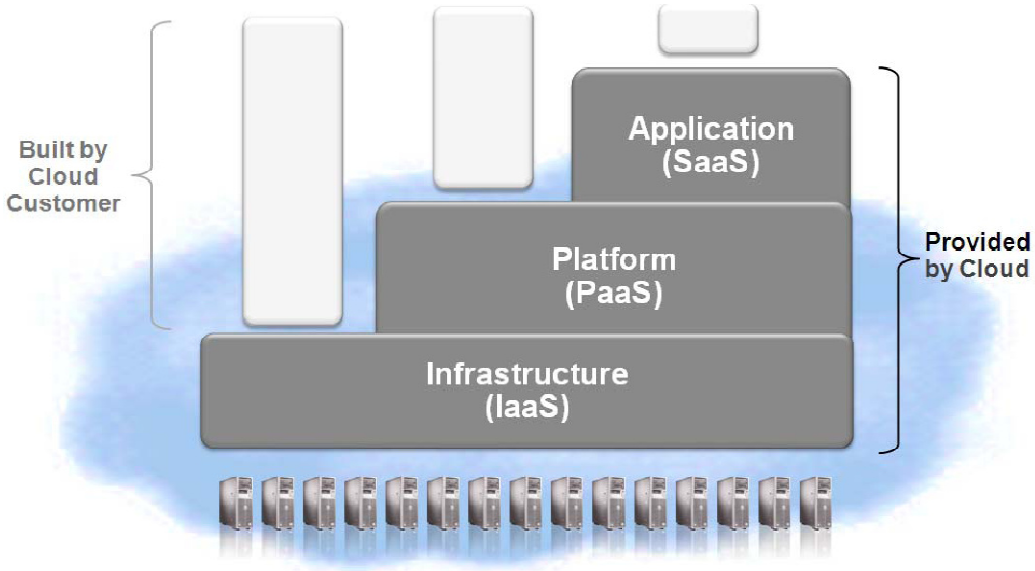
Globální oteplování, snižování emise skleníkových plynů, využívání energie z obnovitelných zdrojů – to všechno jsou v posledních letech velmi významně skloňovaná témata, která mohou nepřímo ovlivnit rychlost adopce služeb v celé oblasti cloud computingu. Především při snižování spotřeby elektrické energie může být jeho role nezanedbatelná. Pokud společnost využívá software jako službu a nemusí provozovat vlastní data centrum, šetří energii nejen za provoz samotných serverů, ale i dalších nezbytných zařízení jako klimatizace, osvětlení či zabezpečení. Navíc například průzkum společnosti McKinsey & Co. zaměřený na využívání serverů v jedné mediální společnosti ukázal, že 1/3 serverů měla míru využití pod 3% a skoro 2/3 serverů byly využívány jen z 10%, což jasně poukazuje na předimenzování a plýtvání zdroji. [NET2009, str.3] Naproti tomu u provozovatelů velkých datových center je v jejich vlastním zájmu využívat zdroje maximálně efektivně. Navíc si mohou dovolit rozsáhlé investice do nejmodernějších úsporných technologií.

Jiný průzkum provedený společností Greenspace se zaměřil na výhody využívání Software-as-a-Service platformy poskytovatele NetSuite oproti klasickému provozu vlastního datového centra. Výsledky ukázaly, že díky využívání svého datového centra společnost NetSuite a její zákaznící dohromady ušetřili během roku 2008 595 milionů kWh elektrické energie, což je roční spotřeba 56 tisíc domácností. Zároveň bylo sníženo množství vypouštěných skleníkových plynů v takovém objemu, který odpovídá znečištění, které během jednoho roku způsobí 77 tisíc automobilů. [NET2009, str.3-4]

2.9 Modely poskytování služeb

Oblast cloud computingu bývá obecně dělena na tři základní modely poskytování služeb. Někteří autoři sice vymezují více vrstev cloud computingu [JOH2008], tato práce se však zaměřuje pouze na tři základní modely uvedené například v [MEL2009, str. 2]:

- Software-as-a-Service (SaaS)
- Platform-as-a-Service (PaaS)
- Infrastructure-as-a-Service (IaaS)



Obr. 2.9 : Různé modely cloud computingu [PIE2009]

Jednotlivé vrstvy nemají přesně vymezené hranice a může docházet k prolínání modelů. Nejstarší a nejznámější je model SaaS, který může nabývat nejrůznějších podob od webového emailu až po robustní CRM nebo ERP aplikace. V posledních letech však především díky virtualizaci dochází k rozvoji dalších modelů, kdy už nejsou jako služba nabízeny pouze aplikace, ale rovnou celá infrastruktura (IaaS) či platformy pro vývoj vlastních aplikací a jejich následný provoz na infrastruktuře poskytovatele (PaaS).

2.9.1 Software-as-a-Service

Tento model bývá často označován také jako aplikační vrstva. Oproti tradičnímu způsobu dodávání software zde odpadá nutnost cokoli instalovat, veškerá funkcionalita aplikace je

dostupná odkudkoliv přes internet pomocí webového prohlížeče. Mnohem jednodušší je také aktualizování a nasazování nových verzí programů, kdy je nutné provést upgrade pouze na jednom centrálním místě, kde aplikace běží.

Poskytovatel služby nemusí být zároveň majitelem infrastruktury, na které aplikace běží. Může využívat infrastrukturu v cloudu jiného poskytovatele. Dochází tedy k vytvoření určité hierarchie poskytovatelů a uživatelů v rámci vrstveného modelu cloud computingu.

Hlavní charakteristiky:

- jedna aplikace je využívána zároveň mnoha uživateli
- aplikace je poskytována přes HTTP protokol
- účtování pay-as-you-go způsobem

Výhody:

- minimální počáteční náklady a náklady na údržbu
- krátký implementační cyklus
- poskytovatel je motivován nabízet co nejlepší služby, jinak nedostane zaplaceno

Možná omezení:

- omezená customizace aplikace
- ztráta kontroly nad daty a funkcionalitou
- spolehlivost poskytovatele

Příklady SaaS:

- Google Apps – balík aplikací sloužící především k podpoře týmové práce
  http://www.google.com/apps/intl/cs/business/index.html
- Salesforce – CRM aplikace
  http://www.salesforce.com/
- IBM LotusLive – sada aplikací pro podporu týmové spolupráce
  http://www-01.ibm.com/software/cz/lotus/category/saas/
- jakýkoliv webmail

2.9.2 Platform-as-a-Service

Tomuto modelu se detailně věnuje kapitola 3.

2.9.3 Infrastructure-as-a-Service

Někdy se také hovoří o tzv. Hardware-as-a-service. V rámci IaaS jsou zákazníkům zprostředkovávány služby nabízející využití výpočetních zdrojů, datových úložišť a komunikačních sítí. Na tuto pronajatou infrastrukturu si pak může uživatel nainstalovat libovolný operační systém či aplikace a následně je využívat pomocí internetu podobně, jako kdyby byly nainstalovány lokálně. Uživatel nemá plnou kontrolu nad nejnižší vrstvou infrastruktury, může pouze v některých případech řídit výběr síťových komponent, jako např. firewall apod. [MEL2009, str. 2] Tento model je tedy schopen v největší míře využívat všech výše popsaných výhod, které cloud computing přináší.

Hlavní charakteristiky:

- centrálně spravovaná infrastruktura
- elasticita – možnost přidávání a odebírání zdrojů
- vývojáři mají plnou kontrolu nad virtuálními stroji poskytovanými na vyžádání

Výhody:

- omezení počátečních nákladů na vybudování infrastruktury
- nižší náklady na údržbu infrastruktury
- možnost řešení výpočetně náročných úkolů

Možná omezení:

- čas potřebný k uvedení na trh je delší než u SaaS a PaaS [TVA2009, str. 29]
- právní aspekty vyžadující, aby byly určité informace uchovávány na území státu, kde společnost sídlí

Příklady IaaS:

- Amazon EC2 - http://aws.amazon.com/ec2/
- GoGrid - http://www.gogrid.com/
- Rackspace Cloud - http://www.rackspacecloud.com/

2.10 Možné překážky

Cloud computing s sebou kromě mnoha výhod přináší i řadu nových otázek, jejichž uspokojivé zodpovězení je nezbytné k jeho přijetí a možnému dalšímu rozvoji. Jedná se hlavně o oblasti jako legislativa, bezpečnost či různé právní aspekty. Se seznamem deseti nejvýznamnějších překážek, které mohou bránit rozšíření cloud computingu, přichází například odborníci z univerzity v Berkeley [ARM2009, str. 14-19]. Zároveň nabízí i svůj pohled na možná řešení těchto překážek. Tato práce zmiňuje pouze několik nejvýznamnějších.

2.10.1 Bezpečnost

Centrální uložení dat v cloudu samozřejmě vyvolává otázku, zda jsou tato data dostatečně chráněna. Pro potencionálního útočníka je soustředění dat na jednom místě výhodnější, než jejich roztříštěnost po celém podniku. To si samozřejmě poskytovatelé cloudu plně uvědomují a zaměstnávají celá oddělení bezpečnostních specialistů. Na zabezpečení dat je kladen veliký důraz. Samozřejmostí je využívání firewallů a různých způsobů šifrování přenosu dat jako např. SSH nebo HTTPS. Schéma všech bezpečnostních mechanismů je možné najít například v „Přehledu bezpečnostních procesů“[AMA2009] jednoho z největších poskytovatelů v rámci cloud computingu, společnosti Amazon.

Je zde tedy velký předpoklad, že úroveň zabezpečení dat dosahovaná poskytovatelem cloudu bude mnohem vyšší, než jsou schopny dosahovat malé a střední společnosti. Ty tak mohou i na poli bezpečnosti v rámci využívání cloud computingu získat. U velkých korporací s rozsáhlými IT odděleními se tato výhoda částečně maže. I při využívání infrastruktury public cloudu těmito společnostmi tak budou patrně důležitá a citlivá data nadále uchovávána lokálně.

2.10.2 Lock-in

Lock-in je označení pro situaci, kdy data či aplikace nemůže být jednoduše přenesena k jinému poskytovateli. Důvodem může být například nutnost použití nestandardního programovacího jazyka či celého API[3] u původního poskytovatele. Každý pokus o přenesení aplikace ke konkurenci se tedy neobejde bez zásahu do kódu aplikace. To je samozřejmě výhodné pro

[3] Application Programming Interface

samotného poskytovatele, ale může způsobit nemalé problémy uživatelům. Účinným lékem na tento problém je stanovení celosvětově uznávaných standardů. Jisté snahy na tomto poli již můžeme sledovat. Existuje například manifest za otevřenost cloudu [OCM2009] či různá fóra jako například Cloud Computing Interoperabitity Forum[4], která si kladou za cíl rozpoutat diskuzi a pomoci tak ke stanovení standardů v rámci cloud computingu. Nicméně tyto snahy jsou teprve ve svých počátcích a standardizace tak zůstává jednou z největších výzev do budoucnosti pro cloud computing.

2.10.3 Legislativa

Hlavním problémem v této oblasti je fakt, že data jsou ukládána do cloudu, ale fyzické umístění těchto dat je uživateli většinou neznámé. Existují však sady zákonů a regulací, které nedovolují uchovávat jisté informace mimo zemi, ve které společnost působí. Příkladem může být The European Union´s Data Protection Directive[5] nebo americký US Patriot Act[6]. Hlubší rozbor legislativních aspektů cloud computingu přesahuje rozsah této práce, nicméně je nutné zmínit, že jednotliví poskytovatelé se snaží těmto zákonům všemožně vyhovět. Například již zmiňovaná společnost Amazon dovoluje výběr umístění pronajímané infrastruktury buď v Evropě, nebo ve dvou lokalitách v Severní Americe.

3 Platform-as-a-Service

Model Platform-as-a-Service tvoří prostřední vrstvu mezi modely SaaS a IaaS v rámci cloud computingu. Zjednodušeně řečeno vytváří virtuální online prostředí pro celý životní cyklus aplikace od jejího vývoje až po následné provozování či její případný prodej, to vše na infrastruktuře poskytovatele. Umožňuje vývojářům soustředit se na samotnou tvorbu aplikace. Ti se tedy nemusí starat o dostupnost, vyvažování zatížení, zálohování systému, patchování, bezpečnost a další infrastrukturní záležitosti.

[4] www.cloudforum.org
[5] www.ico.gov.uk/upload/documents/library/data_protection/detailed_specialist_guides/review_of_eu_dp_directive.pdf
[6] http://epic.org/privacy/terrorism/hr3162.html

Společnost Forrester definuje Platform-as-a-Service takto:

„Externě hostovaná platforma poskytující služby pro vytváření, běh a obsluhu aplikací, zahrnující nástroje pro vytváření, administraci a management, runtime engine, data management engine, bezpečnostní vybavení a nástroje pro správu uživatelů. PaaS je založen na internetových protokolech a zákonitostech.“ [RYM2009, str. 2]

PaaS může v závislosti na jednotlivých konkrétních platformách poskytovatelů představovat všechny způsoby vytváření aplikace od klasického psaní kódu až po interaktivní skládaní funkcionality z malých částí drag and drop formou. Je zde také možnost zakoupení již hotových aplikací a jejich následná úprava za účelem maximálního pokrytí našich požadavků. Vytvářet novou aplikaci lze také skládáním několika jiných již existujících aplikací či jen jejich částí. V takovém případě hovoříme o tzv. *mashupu*.[HIN2008, str. 4]

3.1 Životní cyklus aplikace v rámci PaaS

V současnosti při klasickém způsobu vývoje aplikací se nezřídka kdy stává, že kód těchto aplikací je psán v jednom prostředí, testy probíhají v jiném a samotné nasazení pak ještě v třetím odlišném prostředí. [MIT2008] Budování, konfigurace a správa všech těchto prostředí s sebou samozřejmě přináší další dodatečné náklady, které se pak projeví ve výsledné ceně vytvářené aplikace. Plnohodnotná platforma v rámci PaaS naproti tomu umožňuje průběh celého životního cyklu aplikace od vývoje, přes testování, nasazení až po spravování v jednom a tom samém prostředí, což vede k výraznému snížení nákladů, času potřebnému k uvedení na trh a omezení rizik souvisejících s projektem. Odpadá nutnost instalace serverů a konfigurace vývojového prostředí u každého programátora. Vývojáři se mohou soustředit jen na vytváření software. [LAW2008, str. 14]

3.1.1 Vývoj aplikací

Způsoby vytváření aplikací se u platforem jednotlivých poskytovatelů odlišují. Můžeme pozorovat dva hlavní směry, kterými se nabízené platformy ubírají. První skupina poskytovatelů (např. Bungee Labs, Force.com) spoléhá na vlastní verze jazyků čtvrté generace na úrovni

metadat, které by měly ulehčit vytváření nových aplikací i méně zkušeným programátorům. Ti samí poskytovatelé také nabízí webové IDE pro tyto jazyky. [DAT2010, str. 563]

Na druhé straně platformy společností jako například Google nebo Microsoft se spoléhají na standardní programovací jazyky jako Java, Python, .NET, Ruby atd. Vývoj aplikací neprobíhá přímo na webu, ale většinou pomocí poskytnutého SDK pro standardní vývojová prostředí jako Eclipse nebo Visual Studio. Aplikace je pak vytvářena lokálně na simulované platformě a až později je odeslána na platformu poskytovatele.

3.1.2 Spolupráce

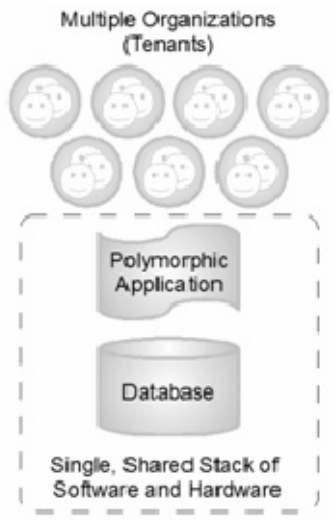
Model PaaS má z principu velký potenciál podporovat spolupráci členů týmů při tvorbě software a usnadnit tak celý proces vývoje. Využívání počítačové sítě k podpoře spolupráce není novinkou, ale u mnoha projektů není využívána dostatečně. [HIN2008, str.9] U PaaS je kód spravován online, což členům týmu usnadňuje přístup k němu a jeho následnou editaci. Platforma by dále měla umožňovat vývojářům vzájemnou komunikaci, sdílení informací a další nástroje podporující společnou práci na projektu. Veškerá tato funkcionalita by měla být přímo součástí dané platformy bez nutnosti instalace dalšího software či nastavování. [MIT2008] Nicméně současné platformy poskytovatelů na trhu PaaS až na pár výjimek (např. Bungee) spolupráci ne zcela podporují. A když už ano, tak pomocí různých podpůrných nástrojů, které jsou produktem téže společnosti (např. Microsoft, Google). [DAT2010, str. 564]

3.2 Architektura platformy

Ačkoliv vytváření aplikací pomocí některé nabízené platformy v rámci PaaS může být velmi snadné a zvládnutelné i méně zkušeným programátorem, musíme si uvědomit, že na straně poskytovatele této platformy naopak musí existovat velmi dobře navržená robustní architektura a řada nástrojů, které toto jednoduché vytváření umožní za současného efektivního využití zdrojů poskytovatele. Technické řešení funkcionality se samozřejmě může a pravděpodobně i bude u jednotlivých platforem různých poskytovatelů odlišovat. Ukázka, jak taková architektura může vypadat, bude v této práci ilustrována na platformě společnosti Salesforce, která se nazývá **Force.com**.

3.2.1 Multitenant aplikace

Aplikace vyvíjené a fungující na platformě Force.com se označují jako multitenant. Tento pojem popisuje přístup k architektuře, kdy aplikace běží pouze v jedné instanci na jednom místě a je sdílena mnoha uživateli, kteří ale žádným způsobem nepoznají, že nevyužívají svou vlastní instanci. Data a individuální nastavení jednoho klienta jsou totiž virtuálně izolována od dat ostatních uživatelů. Výhodou je, že aplikace pro svůj běh vyžaduje pouze jednu sadu hardwarových nástrojů, což poskytovateli velmi usnadňuje administraci.

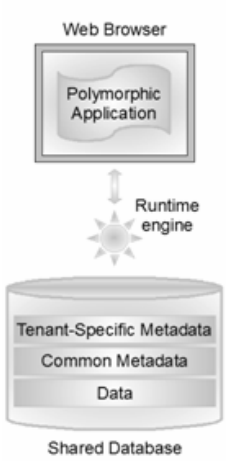


Obr. 3.2.1 : Ukázka multitenant aplikace, která obsluhuje několik uživatelů [WEI2009]

3.2.2 Architektura řízená metadaty

Pro správnou funkcionalitu multitenant aplikace je zapotřebí, aby umožňovala customizaci vzhledu a business logiky jednotlivými uživateli, udržovala jejich data odděleně, zajišťovala bezpečnost dat či umožňovala provádět aktualizace bez dopadu na individuální nastavení uživatelů. Z těchto důvodů Force.com využívá tzv. architekturu řízenou metadaty, která od sebe jasně odděluje runtime engine (kernel), aplikační data, metadata popisující základní funkcionalitu aplikace a metadata odpovídající datům jednotlivých uživatelů. Vše (formuláře, workflow, business logika, reporty atd.), s čím přichází vývojář nebo uživatel u Force.com do styku, je vnitřně reprezentováno metadaty. Ta jsou uchovávána v tzv. Universal Data Dictionary

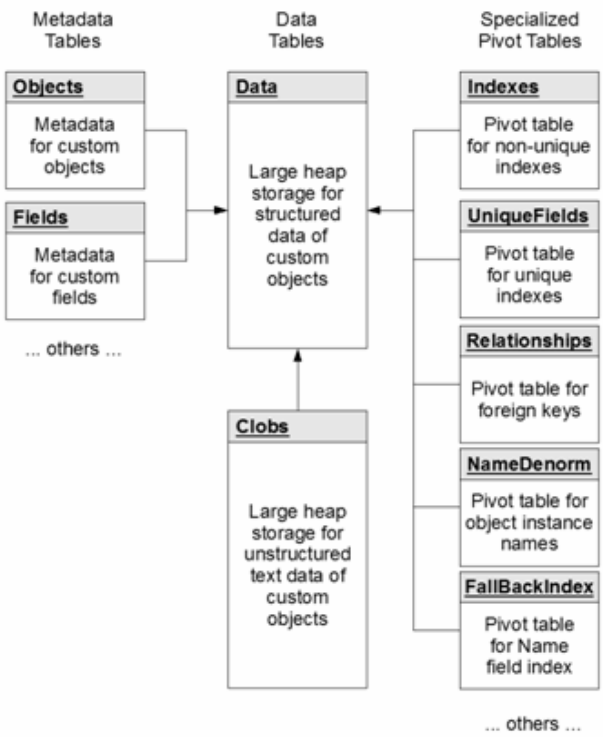
(UDD). Jednotlivé komponenty aplikace jsou pak z metadat dynamicky generovány pomocí runtime enginu.



Obr. 3.2.2 : Schéma architektury řízené metadaty [WEI2009]

3.2.3 Definice dat a ukládání

Data jsou na platformě Force.com ukládána do virtuální databázové struktury, která obsahuje tabulky s metadaty, daty a pivotní tabulky, jak je znázorněno na následujícím obrázku.



Obr. 3.2.3 : Model ukládání dat u Force.com [WEI2009]

Tabulka metadat Objects

Tato tabulka uchovává metadata (ID objektu, název objektu atd.) o vlastních objektech, které si uživatel vytvořil.

Tabulka metadat Fields

Zde jsou uchovávána metadata o polích (sloupcích nebo atributech), které byly definovány pro uživatelem vytvořené objekty.

Tabulka Data

Uchovává data dostupná aplikaci, která jsou popsána pomocí metadat v tabulkách Objects a Fields. Pole mohou obsahovat některý ze standardních strukturovaných datových typů, jako např. text, číslo, datum, checkbox atd. Polím je také možné přiřazovat různá validační pravidla či je nastavit jako povinná.

Tabulka CLOBs

CLOB je zkratkou pro „character large objects“ a slouží pro ukládání dlouhých textových polí o velikosti až 32 000 znaků.

Pivotní tabulka Indexes

Do této tabulky jsou průběžně kopírovány pole označené k indexování z tabulky Data a jejich úkolem je urychlení vyhledávání.

Pivotní tabulka UniqueFields

Tabulka podobná tabulce Indexes, která tentokrát umožňuje zajišťovat jedinečnost jednotlivých polí.

Pivotní tabulka Relationship

Pomocí této tabulky, jak už název napovídá, je možné definovat vztahy (referenční integritu) mezi jednotlivými objekty aplikace.

Další pomocné tabulky jako například **NameDenorm** či **FallBackIndex** slouží většinou k podpoře vyhledávání.

3.2.4 APEX

APEX je označení pro silně typový objektově orientovaný procedurální programovací jazyk, který Force.com využívá.[WEI2009 str.894] Svou syntaxí se podobá Javě a umožňuje programátorům definovat dodatečnou business logiku, operace na manipulaci s daty či například operace na kontrolu transakcí. Více informací včetně ukázek syntaxe je možné nalézt například v dokumentaci platformy Force.com [SAL2007].

Force.com průběžně monitoruje vykonávané operace pomocí nejrůznějších nástrojů a předchází tak škodlivé nebo neúmyslné monopolizaci sdílených zdrojů. Pokud je dosaženo určitých limitů nebo je operace označena za příliš výpočetně náročnou, aplikace může skončit runtime výjimkou. Force.com se snaží tímto způsobem donutit vývojáře k psaní efektivního kódu. S tím souvisí i požadavek na pokrytí alespoň 75% kódu testy a jejich kontrola před tím, než je aplikace napsaná v APEXu spuštěna v ostrém provozu.

Podkapitoly 3.2.1 – 3.2.4 popisují základní pilíře skrývající se za platformou Force.com společnosti Salesforce. Pro bližší pochopení vnitřních mechanismů a detailnější popis využitých technologií je vhodné nastudovat například článek popisující podrobně architekturu platformy, jehož autorem je CTO společnosti Salesforce C. Weissman [WEI2009], anebo navštívit přímo webové stránky s dokumentací platform Force.com [DOC2010].

3.3 Ekonomické přínosy vývoje aplikací pomocí PaaS

Při hodnocení ekonomických přínosů, které přináší vývoj aplikací v rámci PaaS oproti klasickému způsobu, se opět zaměříme na platformu Force.com. Proč právě na ni? Tato platforma byla uvedena do komerčního provozu již v roce 2007 jako jeden z prvních projektů PaaS vůbec. Oproti jiným velkým hráčům na trhu jako například Googlu nebo Microsoftu s jejich platformami, které se v současnosti nachází teprve v raných fázích uvedení do provozu, je tedy u Force.com již možné sledovat ekonomické výsledky organizací, které se rozhodly přesunout své business aplikace právě na platformu společnosti Salesforce. Force.com v současnosti využívá téměř 60 tis. organizací, které zde provozují dohromady více než 100 tis. aplikací. [SAL2010] Na téma ekonomických přínosů využívání této platformy bylo mezi organizacemi provedeno několik průzkumů, například [IDC2009] nebo [NUC2009].

Společnosti oslovené v uvedených průzkumech byly schopny vyvinout novou aplikaci pomocí Force.com průměrně 4,9 krát rychleji než klasickým způsobem v prostředí Javy nebo .NET. Důvodů, proč je vývoj aplikací na Force.com rychlejší, můžeme najít hned několik:

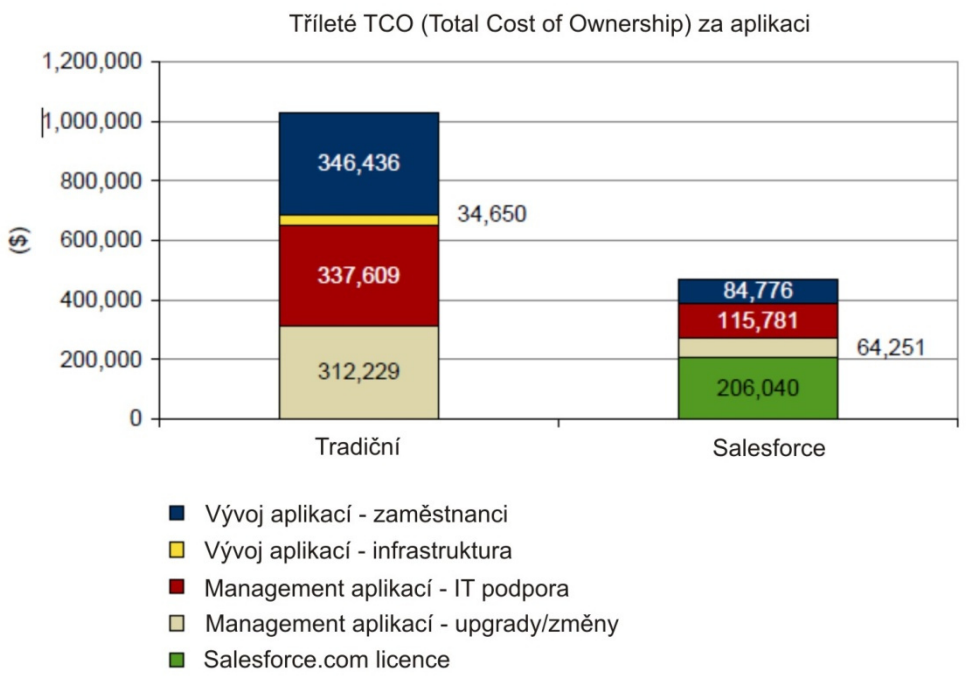
- vlastní objekty a databáze mohou být vytvořeny velmi rychle a znovu-použity v jiných projektech
- možnost rychlé konfigurace administrátorských nástrojů a uživatelského rozhraní; odpadá tedy nutnost vytváření vlastních
- možnost rychlého nastavení pravidel, formulářů a procesů k podpoře vlastních workflow
- platforma je již před-testována; to se týká databází, bezpečnostních protokolů atd., vývojáři tak nemusí dělat tolik strategických technologických rozhodnutí a nemusí strávit tolik času testováním
- nasazování aplikace je v podstatě pouze záležitostí „stisknutí tlačítka“

Podle průzkumu společnosti IDC [IDC2009, str. 10-11] byly zkoumané společnosti schopny snížit mzdové náklady na vývoj a nasazení jedné aplikace o 76%. Se zjištěním, že vlastní vyvíjení aplikací je rychlejší a levnější, zároveň u sledovaných organizací vzrostl počet plánovaných vlastních projektů.

3.3.1 Nasazení a provoz aplikací

Výhody nepřináší pouze samotný vývoj software na platformě v rámci PaaS, ale i následný provoz výsledných aplikací. Podle již výše uvedeného průzkumu [IDC2009] zaměřeného na platformu Force.com, počet volání helpdesku poklesl o třetinu a problémy byly řešeny téměř o polovinu rychleji než u implementace klasických aplikací. To dohromady vedlo k redukci aktivit pracovníků helpdesku o 63%. Nízká chybovost a vysoká spolehlivost aplikací rovněž napomohla k výraznému snížení doby odstávek v prvním měsíci (o 82%) i roce (o 97%) běhu aplikace. Dále se také výrazně zmenšila doba potřebná pro nezbytný update a správu aplikace.

Pokud zkombinujeme výhody vývoje aplikací na platformě Force.com s jejich výhodami za první 3 roky užívání, tak podle průzkumu došlo u sledovaných společností k poklesu celkových nákladů o 54% oproti klasickému vývoji in-house, což znázorňuje následující graf.



Obr. 3.3.1 :Celkové náklady na aplikaci, převzato a upraveno z [IDC2009]

3.4 Možné obavy

Každá změna je většinou spojena s obavami, zda přinese obrat k lepšímu či k horšímu. Kapitoly výše vymezují převážně přínosy a výhody, které vyvíjení aplikací v rámci PaaS organizacím přináší. Pro úplnost je však potřeba uvést i několik bodů charakterizujících určitá znepokojení, kterým může organizace při vyvíjení a provozu aplikací na platformě poskytovatele čelit.
V různých článcích a průzkumech jsou autory zmiňovány tyto body:

- nutnost naučit se vyvíjet a spravovat aplikace pomocí webových nástrojů poskytovatele, případně potřeba osvojení si nového programovacího jazyka [LAW2008, str. 15]
- hrozba lock-in (viz kapitola 2.10.2) [MOO2009, str. 6]
- možná technická nezralost platformy; model PaaS je stále ve své rané fázi a neustále se vyvíjí [MOO2009, str. 7]
- existence určitých limitů na zdroje; nemusí platit globálně pro všechny platformy v rámci PaaS, př.: u Force.com existují určité nástroje („hlídači“), kteří zajišťují stabilitu multitenant aplikace pomocí různých limitů (např. na vykonávání skriptů). Force.com tedy nemusí být ideální prostředí pro běh výpočetně velmi náročných aplikací [IDC2009, str. 15]

- vyvíjení aplikací online může být rizikové z hlediska výpadku připojení k internetu [LAW2008, str. 15]

4 Přehled dostupných platforem v rámci PaaS

Trh s produkty nabízenými v rámci modelu Platform-as-a-Service je poměrně mladý a v současnosti velmi dynamicky se rozvíjející. K již zaběhlým a několik let fungujícím projektům se v nedávné době připojili i světoví softwaroví giganti Google a Microsoft, kteří představili a spustili své řešení platforem. Nabízené služby jednotlivých poskytovatelů se neomezují pouze na oblast PaaS, ale velmi často dochází k prolínání s dalšími dvěma základními modely cloud computingu - SaaS a IaaS.

Oblast PaaS je velmi široká a nabízená funkcionalita jednotlivých nabízených platforem se od sebe může značně odlišovat. Poskytovatele můžeme velmi hrubě rozdělit do dvou skupin podle způsobu přístupu k vývoji aplikací. První a početnější skupina zahrnuje platformy, které umožňují vytváření aplikací v prohlížeči za pomoci online vývojového prostředí drag and drop formou. Psaní kódu není nutné a nebo jen v minimální míře. Většinou jsou tímto způsobem vytvářeny nejrůznější databázové a business aplikace. Na opačné straně stojí skupina platforem, pro které jsou aplikace vyvíjeny klasicky offline psaním kódu určitého programovacího jazyka, a až hotová aplikace je přesunuta do cloudu poskytovatele, kde potom běží. Tento způsob je tedy blíže k modelu IaaS s tím rozdílem, že aplikace jsou vyvíjeny pro danou platformu, v daném jazyce a musí respektovat určitá pravidla a omezení specifická pro vybraného poskytovatele. Samostatnou skupinu pak ještě tvoří platformy zaměřené na tvorbu a spravování webových stránek. Tato práce se zaměřuje na všechny výše uvedené skupiny a v rámci nich na celé spektrum dostupných řešení.

Platformy pro detailnější analýzu byly převážně vybrány na základě obsahu dokumentu společnosti Gartner, který se zabývá platformami v rámci cloud computingu.[NAT2009] Použit byl skutečně jenom obsah se seznamem platforem, protože přístup k plné verzi dokumentu je zpoplatněn.

4.1 Platformy pro vývoj aplikací v internetovém prohlížeči

Platformy v této kategorii umožňují vytvářet webové aplikace bez nutnosti psaní kódu. Nabízí nástroje pro tvorbu formulářů drag and drop formou, definice workflow a business procesů či vytváření databází. Není tedy nezbytné nic instalovat na lokální počítač, i když některé platformy (např. Force.com) umožňují stáhnutí IDE a práci offline. Webová prostředí hojně využívají pro vytváření aplikací AJAX[7] a jiné nejmodernější webové technologie. Dalším společným znakem je podpora nejrůznějších podrobných statistik a grafů jak o samotných datech, tak i o využívání celé aplikace uživateli. Většina zástupců v této skupině nabízí vytvoření jedné či více aplikací pro omezený počet uživatelů zcela zdarma.

4.1.1 ZOHO Creator (creator.zoho.com)

ZOHO Creator je jednoduché prostředí pro vytváření databázových aplikací, jakými mohou být například různé dotazníky, registrační formuláře či evidence. Nabízí zajímavou alternativu ke klasickým kancelářským softwarům jako MS Excel a MS Access s tím, že data jsou pomocí internetu dostupná kdykoliv a odkudkoliv. Umožňuje vytvářet aplikace zcela od začátku a nebo na základě naimportovaných dat. Podporované formáty souborů pro import jsou .xls, .mdb, .csv a .tsv. Základ každé aplikace tvoří formuláře a pohledy. Formuláře slouží k získávání dat a jsou vytvářeny jednoduše skládáním jednotlivých komponent drag and drop formou. Podporují více než 20 různých typů datových polí. Pohledy (Views) slouží naopak k zobrazování dat uložených v databázi. Business logika je aplikaci dodávána pomocí Deluge skriptu. Tento skript se generuje automaticky v závislosti na tom, jak opět drag and drop formou myší přetahujeme a nastavujeme komponenty workflow. Jednotlivé části skriptu se spouští na základě událostí (např. On Success, On User Input, On Add atd.).

ZOHO má integrovánu podporu REST[8] a RPC[9], pomocí kterých mohou aplikace zde vytvořené komunikovat a vyměňovat si data s jinými webovými aplikacemi. Uživatelé tak mají možnost si vytvořit libovolnou aplikaci pomocí PHP, JSP, ASP, Perlu či jiného programovacího jazyka, zatímco jejich data budou ukládána v ZOHO.

---

[7] Asynchronous JavaScript and XML
[8] Representational State Transfer
[9] Remote procedure call

Velmi podobnou funkcionalitu pro tvorbu jednodušších databázových aplikací stejně jako ZOHO nabízí například platformy společností Caspio (www.caspio.com), DabbleDB (www.dabbledb.com) či TeamDesk (www.teamadesk.net).

4.1.2 Qrimp (www.qrimp.com)

Qrimp je PaaS řešení pro vytváření databázových aplikací, které oproti konkurenčním platformám s podobným zaměřením přináší zajímavou funkcionalitu v několika ohledech. Nejrychlejší způsob, jak sestavit fungující aplikaci, je s pomocí tzv. modulů. Moduly jsou předpřipravené tabulky s možností editace zaměřené na nejrůznější témata, jako například Zákazníci, Účty, Videa, Kontakty atd. Velice jednoduchý je také import dat z Excelu, kdy si vystačíme pouze s příkazy „kopírovat“ a „vložit“ a Qrimp už naše data sám umístí do tabulek. Data mohou být následně prohlížena a editována v několika různých pohledech včetně kalendářového či stromového. Další zajímavou nabízenou vlastností je možnost práce offline a následná synchronizace dat tak, aby byla zachována jejich konzistence. Aplikace vytvořená v Qrimpu nemusí běžet pouze ve veřejném cloudu, ale existuje i možnost provozu na vlastním serveru za vlastním firewallem, pokud je to z bezpečnostních či jiných důvodů vyžadováno. Qrimp je vhodná platforma pro jednoduché až středně složité databázové aplikace a mezi její hlavní přednosti patří přehlednost a jednoduchost. Navíc první dva měsíce využívání platformy jsou u všech nabízených edic zcela zdarma.

4.1.3 WaveMaker (www.wavemaker.com)

Wavemaker je open source platforma pro vývoj software, která využívá otevřené standardy a výrazně tak snižuje hrozbu lock-in[10]. Skládá se ze dvou hlavních komponent: Studia a Serveru. Wavemaker Studio slouží k vytváření aplikací drag and drop formou přímo v prohlížeči a díky jeho interaktivně můžeme hovořit o tzv. WYSIWYG[11] přístupu. Naproti tomu Server slouží k nasazování již hotových aplikací do standardního Java prostředí. Aplikace vytvořené na této platformě jsou založeny na open source komponentách Javy a Javascriptu, jako například Dojo, Tomcat nebo Hibernate. Pro usnadnění autorizace uživatelů aplikací Wavemaker podporuje

[10] Vysvětlení, co je to lock-in, se nachází v kapitole 2.10.2.
[11] What You See Is What You Get – verze zobrazená při editaci je totožná s verzí výslednou

standard JOSSO[12]. Samozřejmostí je také podpora prostředků jako REST, SOAP[13] nebo RSS pro propojování více webových aplikací dohromady. Wavemaker je v současnosti dostupný ve třech verzích – Community, Enterprise a Cloud. Aplikace vytvořené v Community edici jsou vydávané pod Apache licencí. V rámci PaaS nás však bude zajímat hlavně verze Cloud, která v současnosti stále nese označení beta. Díky otevřenosti platformy může výsledná aplikace běžet v jakémkoli cloudu (primárně je podporován Amazon EC2) a je bez problémů přenositelná. Zajímavou vlastností je také možnost aplikaci kdykoli převést z verze pro cloud na klasickou open source verzi.

4.1.4 Wolf Frameworks (www.wolframeworks.com)

Wolf je společnost založená v roce 2006 v Indii a poskytující prostředí pro vývoj a provoz SaaS business aplikací bez nutnosti psaní technického kódu. Pro vyvíjení aplikací platforma využívá technologie jako AJAX, XML či .NET. Tvorba softwaru se omezuje na definování business procesů a pravidel „point and click“ formou, kdy v závislosti na umístění kurzoru myši na obrazovce jsou vývojáři zpřístupněny různé možnosti. Aplikace nemusí být vytvářeny úplně od začátku, ale je možné využít některé již hotové řešení a editací procesů ho pouze upravit tak, aby maximálně vyhovovalo našim konkrétním požadavkům. [DNA2010] Platforma dále podporuje jednoduchý import/export dat do Excelu či Wordu. Bezpečnost je zajišťována přístupem k aplikacím přes 128-bitový šifrovaný protokol SSL. V současnosti je dostupných šest licenčních verzí, jejichž ceny se pohybují v rozmezí 7-10 amerických dolarů za uživatele na měsíc, základní edice je zdarma. Wolf představuje ideální řešení pro zákazníky, kteří hledají prostředí pro rychlý vývoj malých až středně velkých business aplikací, aniž by museli napsat jedinou řádku kódu.

4.1.5 Longjump (www.longjump.com)

Longjump je dalším klasickým zástupcem z řad poskytovatelů nabízejících platformu pro vývoj a provoz multitenant business aplikací. Kromě rychlého vytváření aplikací pomocí předem definovaných objektů drag and drop způsobem nabízí možnost zajištění hlubší funkcionality psaním vlastního kódu v Javě nebo Javascriptu díky plug-inu do IDE Eclipse. Propojitelnost s jinými aplikacemi je umožněna díky podpoře standardních protokolů webových služeb REST a

[12] Java Open Single Sign On
[13] Simple Object Access Protocol

SOAP. Longjump je nabízen ve třech základních edicích – Bronze, Silver a Gold. Cenová politika se liší od konkurence tím, že jednotlivé edice od sebe nejsou diverzifikovány počtem uživatelů, ale počtem objektů a velikostí úložiště pro jednoho uživatele. Longjump tedy může být ideální řešení pro situaci, kdy potřebujeme provozovat relativně malou aplikaci pro velký počet uživatelů. Platforma nabízí velkou flexibilitu i v možnostech nasazení. Výsledná aplikace může běžet za podnikovým firewallem, v soukromém data centru a nebo na infrastruktuře libovolného IaaS poskytovatele.

4.1.6 OrangeScape (www.orangescape.com)

OrangeScape je indická společnost, která v rámci PaaS přichází se svým webovým prostředím pro vývoj převážně business aplikací, které se vzhledem a funkcionalitou velmi nápadně podobá tabulkovému procesoru MS Excel. To může řadě uživatelů usnadnit první kroky práce s touto platformou, když se budou pohybovat v pro ně známém prostředí. OrangeScape Studio obsahuje čtyři návrhové perspektivy: modelovou, formulářovou, procesní a akční. Každá z nich poskytuje jiný pohled na vytvářenou aplikaci. V modelové perspektivě se definují základní datové prvky a business pravidla zatímco formulářová slouží k vytváření stránek skládáním jednotlivých buněk, sekcí nebo formulářů. Procesní pohled umožňuje vizuálně definovat workflow procesy. Pomocí akční perspektivy můžeme zase kombinovat různé příkazy do řetězců a použít je například jako emailová upozornění.

Hotový software může být umístěn na „tržiště“, odkud je dostupný pro potencionální klienty. Následně může být provozován v cloudu jako SaaS aplikace a nebo může být stáhnut a fungovat lokálně jako klasický on-premise software. Cloud edice se nicméně nachází zatím pouze ve fázi testování a nejasná je i budoucí cenová politika.

4.1.7 WorkXpress (www.workxpress.com)

Další z řady dostupných platforem je WorkXpress. Ačkoli se chlubí tím, že je to jediná platforma využívající programovací jazyk 5. generace (5GL PaaS), ve skutečnosti se svou nabízenou funkcionalitou nikterak výrazně neliší od svých konkurentů ve stejné kategorii. Opět se jedná o webové prostředí, kde jsou aplikace budovány drag and drop formou. WorkXpress si velmi zakládá na vizuální stránce. V průběhu vývoje nás tedy doprovází různobarevné ikonky a

obrázky, které mají za cíl zpřehlednit celý cyklus vývoje. Barevně je odlišeno i pět základních stavebních kamenů, z kterých aplikace sestává. Konkrétně to jsou Položky, Pole, Vztahy, Formuláře a Akce. Již z jejich názvů je patrné, k čemu bude který blok sloužit. Uživatelům je dále k dispozici „obchod“ s již hotovými aplikacemi, které je možno si podle potřeby přizpůsobovat. WorkXpress tímto způsobem nabízí jak jednoduchá softwarová řešení konkrétních oblastí, tak i komplexní aplikace typu CRM a ERP. Vytváření aplikace na této platformě je zdarma, platí se až za využívání zdrojů při provozu. Výsledné aplikace mohou běžet v soukromém cloudu či na infrastruktuře některého z poskytovatelů v rámci IaaS, kteří jsou partnery WorkXpressu (Amazon, GoGrid).

WorkXpress je tedy platforma pro jednoduchý vývoj převážně business aplikací bez nutnosti psaní kódu, která vyniká především svým grafickým zpracováním.

4.1.8 Bungee Connect (www.bungeeconnect.com)

Platforma Bungee je dalším příkladem řešení pro vytváření aplikací online přímo prostřednictvím webového prohlížeče. Tentokrát můžeme využít prostředí založené na známém IDE Eclipse. Většina funkcionality aplikace se nastavuje za použití drag and drop nástrojů, ale k dispozici je i speciální programovací jazyk nazývaný Bungee Sky, který se svou syntaxí podobá jazyku C#. V čem Bungee vyniká oproti ostatním podobným produktům, je podpora spolupráce vývojářů.[DAT2010, str. 564] Při vývoji aplikace není potřeba pracovat ještě se vzdáleným úložištěm, kde jsou spravovány jednotlivé verze. Ke spuštění týmového projektu stačí jednoduše vytvořit tzv. DesignGroup a pozvat do ní ostatní členy vývojářského týmu. Tento přístup pak umožní sdílení kódu bez jakéhokoli dalšího nastavování.

Problémem platformy Bungee ale pravděpodobně bude její nedostatečná zralost, jelikož se v současnosti čeká na vydání nové beta verze, která bude samozřejmě potřebovat nějaký čas na vyladění.

4.1.9 Force.com (www.force.com)

Force.com je platforma pro vývoj business aplikací od americké společnosti Salesforce, která se proslavila především díky svému CRM řešení nabízenému jako SaaS. Na rozdíl od ostatních platforem uvedených výše v této kategorii umožňuje vyvíjení aplikací nejen pomocí

jednoduchého vytváření objektů a business pravidel v prostředí webového prohlížeče, ale i klasicky za použití proprietárního jazyka APEX[14] v integrovaném vývojovém prostředí Eclipse. Obě dvě varianty mají samozřejmě své výhody i nevýhody. Použití tzv. „builderu“ v prohlížeči je jednoduché a uživatelsky přívětivé. Na druhou stranu ale vyžaduje být neustále připojen k internetu a vytváření velkého počtu objektů s mnoha poli se může změnit v dlouhé neustálé klikání na tlačítko „next“. Pro účely rozsáhlých změn či právě vytváření mnoha podobných objektů bude výhodnější využít Force.com IDE plug-in. S jeho pomocí můžeme psát kód i v režimu offline a celkově umožňuje definovat hlubší funkcionalitu. Nevýhody tohoto způsobu jsou naopak nutnost instalace plug-inu a Eclipse, menší přehlednost a samozřejmě nutná znalost technik programování.

Na stránkách platformy Force.com můžeme najít rozsáhlou a velice dobře zpracovanou dokumentaci, řadu textových i filmových tutoriálů, které potencionálním uživatelům umožní pochopit fungování celé platformy a aktivním vývojářům pomohou vyřešit případné problémy.

Pro naši aplikaci můžeme využít automaticky generované uživatelské rozhraní. K vytvoření vlastního slouží nástroj zvaný Visualforce, který poskytuje značkovací jazyk podobný HTML. Kód je dále možné doplnit o HTML, JavaScript nebo Flash. Stránky ve Visualforce dále podporují například pravidla pro validaci dat, kaskádové styly nebo technologii AJAX. Visualforce umožňuje dokonce vyvíjet stránky přizpůsobené přímo pro některá mobilní zařízení jako Iphone nebo Blackberry.

Uživatelům je k dispozici v katalogu AppExchange více než 800 funkčních aplikací rozdělených podle několika kritérií. Force.com je dostupný ve třech edicích, z nichž základní je zdarma a umožňuje vytvořit jednu aplikaci až pro 100 klientů. Nevýhodou je, že kvůli využívání proprietárních technologií na této platformě existuje poměrně vysoké riziko lock-in.

Dalšími zástupci, které svou funkcionalitou spadají do této kategorie, jsou platformy společností Iceberg (www.geticeberg.com) a Rollbase (www.rollbase.com).

Přehled všech výše zmíněných platforem a jejich základních charakteristik se nachází v tabulce v příloze A.

[14] Popis APEXu se nachází v kapitole 3.2.4.

4.2 Platformy pro vývoj aplikací lokálně

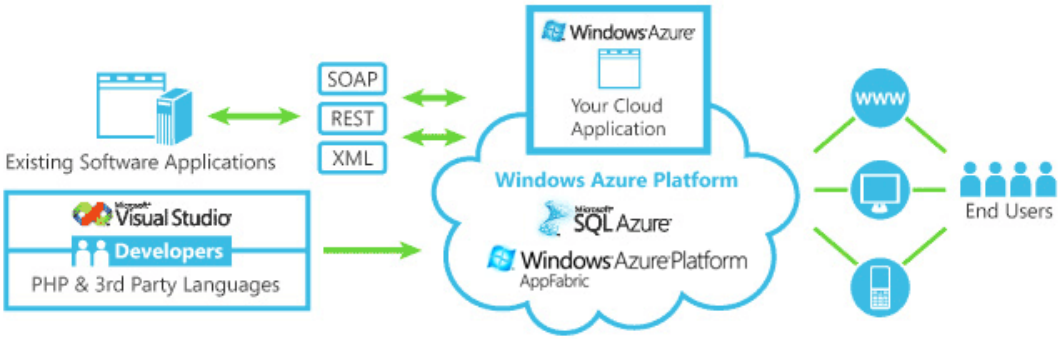
4.2.1 Microsoft Azure (www.microsoft.com/windowsazure)

Azure představuje první velký počin společnosti Microsoft na poli cloud computingu. Platforma byla spuštěna v roce 2009 ve zkušební verzi (Community Technology Preview) a od února 2010 je postupně uváděna do provozu její komerční verze v jednotlivých zemích. Azure poskytuje širokou sadu internetových služeb včetně nového stejnojmenného operačního systému, které běží v datových centrech Microsoftu. Cílem je umožnit vývoj, provoz a spravování různých druhů aplikací, které mohou být umístěny jak v cloudu, tak i lokálně.

Platforma Azure se skládá ze 3 základních komponent:

- Windows Azure
- SQL Azure
- Windows Azure platform AppFabric

Windows Azure vytváří prostředí pro běh aplikací a ukládání jejich dat. SQL Azure je založen na SQL Serveru a poskytuje datové služby v cloudu. AppFabric naproti tomu slouží k propojování jednotlivých aplikací. Detailnější popis fungování celé platformy Azure je možné nalézt například v článku Davida Chappella [CHA2009].



Obr. 4.2.1: Schéma fungování platformy Azure,
zdroj: www.microsoft.com/windowsazure/products/

Samotný vývoj aplikací pro platformu Azure probíhá v integrovaném vývojovém prostředí Microsoft Visual Studio, do kterého je nezbytné doinstalovat nástroj, který bude lokálně simulovat prostředí cloudu. Kód je psán v některém programovacím jazyku pro vývoj .NET aplikací jako například C#. Nicméně Microsoft nabízí ke stažení i SDK[15] pro Java, PHP a Ruby vývojáře. Platforma také podporuje standardy a protokoly jako například SOAP, REST nebo XML.

4.2.2 Google App Engine (code.google.com)

Trh s platformami pro vývoj webových aplikací se v nedávné době rozhodl svým řešením rozšířit i americký softwarový gigant Google. Jeho platforma App Engine byla do zkušebního provozu spuštěna v roce 2008 a zpočátku umožňovala psaní aplikací pouze v jazyku Python. V současnosti už je možné programovat i v Javě a dalších jazycích běžících na JVM[16] (JRuby, Rhino, Scala atd.). Pro vývoj v Javě Google poskytuje plug-in do vývojového prostředí Eclipse, ze kterého je pak možné výslednou aplikaci jednoduše nahrát na jejich infrastrukturu. Pracujeme tedy lokálně offline a webové prostředí platformy je simulováno pomocí SDK.

App Engine obsahuje řadu pravidel, která musí být pro úspěšný vývoj a provoz aplikací dodržována a každý vývojář by se s nimi měl předem seznámit. Příkladem takového pravidla je například filesystem pouze pro čtení, omezení na velikost souborů či zákaz jakýchkoliv systémových volání[17]. Zvláštností je také používání databáze Datastore, která je založena na databázovém systému BigTable společnosti Google. Datastore není relační databáze, nemá žádné schéma a nepodporuje ani SQL, místo kterého využívá speciální dotazovací jazyk GQL[18]. GQL však umožňuje vykonávat příkaz Select pouze nad jednou tabulkou a není zde možnost využít výběr dat z více tabulek příkazem Join. Na to je potřeba myslet již při návrhu aplikace a strukturu našich dat těmto omezením přizpůsobit. Nezbytné patrně bude nenechat se ovlivňovat zažitými stereotypy z relačních databází, data částečně denormalizovat či přenést odpovědnost za konzistenci dat z databáze na aplikaci.[HOL2009] Google se za tato omezení nikterak neomlouvá a dokonce slovy produktového manažera pro App Engine Chrise Beckmanna tvrdí, že pokud se jim uživatel přizpůsobí, dosáhne na oplátku velmi vysokého výkonu.[BAB2009]

[15] Software Development Kit
[16] Java Virtual Machine
[17] Kompletní seznam omezení je uveden zde: http://code.google.com/intl/cs/appengine/docs/quotas.html
[18] Google Query Language

Vytváření aplikací na platformě App Engine je zdarma do určitých limitů. Na jeden účet vývojáře může být registrováno až 10 aplikací. Každá aplikace může využívat až 500MB prostoru a počet zobrazených stránek je omezen na 5 milionů měsíčně. Dodatečné zdroje je možné kdykoli přidávat a jejich aktuální ceník je uveden na webových stránkách platformy[19]. Existuje dokonce možnost, pro každý ze zdrojů nastavit individuální limity a výši maximálního rozpočtu.

4.2.3 Stax Networks (www.stax.net)

Stax zajišťuje prostředí pro management a provoz Java aplikací. Ty ve skutečnosti běží na infrastruktuře EC2 od Amazonu. Aplikace můžeme vytvářet pomocí Stax SDK lokálně v libovolném vývojovém prostředí, i když nové projekty jsou primárně generovány pro IDE Eclipse. Platforma podporuje řadu nejrůznějších oblíbených Java technologií jako například JRuby on Rails, Jython nebo Adobe Flex. Celý projekt se prozatím nachází v beta verzi.

4.2.4 Heroku (www.heroku.com), Engine Yard (www.engineyard.com)

Heroku a Engine Yard jsou platformy určené pro vývojáře webových aplikací využívajících framework Ruby on Rails a programovací jazyk Ruby. Spíše než na samotný vývoj se zaměřují na hosting těchto aplikací. Plusem platformy Heroku je možnost sestavit si individuální balík potřebných zdrojů a různých add-onů a tím pádem i vlastní měsíční platební plán.

Platformy uvedené v této kapitole shrnuje tabulka v příloze B.

4.3 Platformy pro tvorbu webových stránek a portálů

Platformy popsané výše v kapitolách 4.1 a 4.2 jsou převážně určené pro tvorbu intranetových aplikací. Naproti tomu platformy spadající do této kategorie se svou nabízenou funkcionalitou primárně zaměřují na vývoj „frontendových“ aplikací, jakými jsou například běžné weby, různé portály nebo e-shopy.

[19] http://code.google.com/intl/cs/appengine/docs/billing.html

4.3.1 Sitemasher (www.sitemasher.com)

Sitemasher je platforma určená pro webové vývojáře, kterým by měla pomoci vytvářet a spravovat webové stránky rychleji a s menšími náklady. Zajímavostí je, že je vybudována na jiné PaaS platformě, konkrétně na Microsoft Azure. Stránky jsou budovány online v prohlížeči pomocí Studia. Vytváření probíhá převážně drag and drop WYSIWYG formou, ale je možné přidávat i vlastní kód HTML, CSS nebo JavaScriptu. Návrh stránek usnadňují i jednoduché nástroje pro tvorbu formulářů, které jsou ihned propojovány s automaticky generovanou databází. Konečný zákazník, pro kterého byl webový portál budovaný, má podle předem definovaných práv možnost provádět úpravy obsahu stránek jednoduchým způsobem podobným textovému editoru. Platforma také automaticky generuje širokou škálu nejrůznějších statistik a obsahuje i základní nástroje pro SEO[20] optimalizaci.

4.3.2 Cloudia CMS (www.cloudia.cz)

Cloudia CMS je nový zajímavý český projekt, který se svou nabízenou funkcionalitou, jak už název napovídá, přibližuje systémům pro správu obsahu[21]. Na rozdíl od různých běžně dostupných CMS řešení je však zaměřen na správu velmi rozsáhlých (tisíce až stovky tisíc sekcí, stránek, produktů atd.) webových portálů. Příklady projektů vhodných pro tento systém mohou být například mnohojazyčná korporátní prezentace, informační portál pro publikování a šíření zpráv, korporátní intranet nebo extranet, redakční systém či různé e-shopy a rezervační systémy. Základní částí Cloudie je webová aplikace sloužící jako interaktivní administrační rozhraní. Ta mimo jiné nabízí drag and drop ovládání, řadu nápověd a náhledů nebo úpravu dat ve stylu MS Excel. Cloudia také poskytuje různé možnosti rozšiřování funkcionality, které ale většinou vyžadují znalosti programovacích postupů. Součástí systému jsou i SEO nástroje, které základní technické záležitosti jako titulky stránek či vyplňování doplňkových HTML parametrů řeší automaticky.

[20] Search Engine Optimization
[21] CMS – Content Management System – Systém pro správu obsahu

4.3.3 Homestead (www.homestead.com)

Homestead nabízí možnost jednoduchého vytváření webových stránek a e-shopů drag and drop formou. Weby jsou budovány upravováním vzorových stránek, kterých je k dispozici přes dva tisíce a jsou rozděleny do mnoha kategorií. Grafika těchto vzorů může být libovolně upravována například pomocí galerie s více než 250 tisíci obrázky. Vytvoření e-shopu je také velice jednoduché, stačí do již existujících stránek přidat patřičný modul. Homestead zároveň slouží i pro hostování vytvořeného webu. Jednotlivé nabízené edice se od sebe odlišují velikostí úložiště, počtem stránek a další funkcionalitou. Ceny začínají na 4.99$ za měsíc, vytvoření stránek a prvních 30 dní provozu je zdarma. Homestead ocení především ti uživatelé, kteří požadují dobře graficky vypadající stránky ve velmi krátkém čase a netrvají na originalitě řešení.

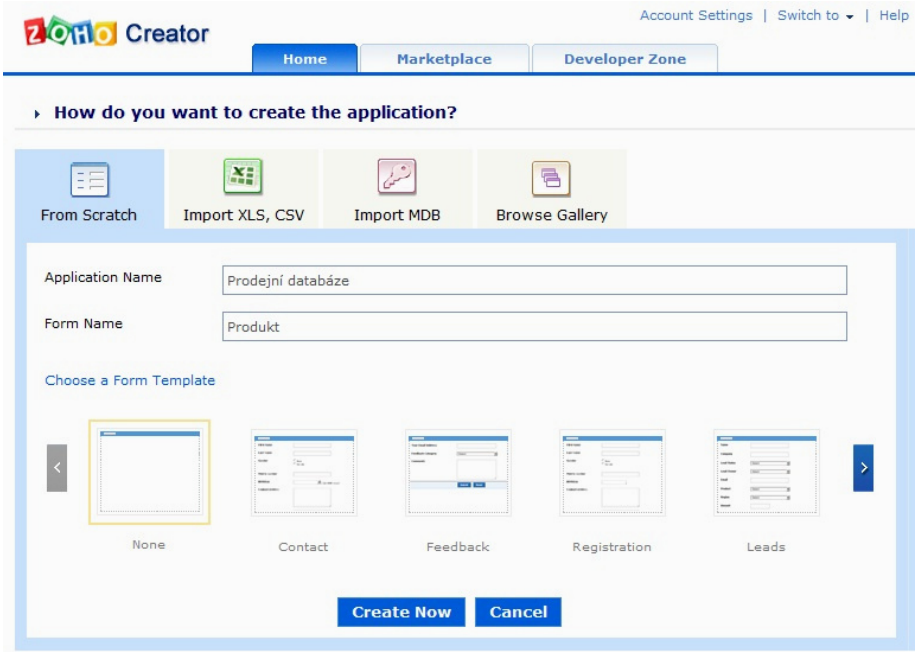
5 Vytvoření jednoduché aplikace

V této kapitole demonstrujeme proces vytvoření jednoduché databázové aplikace na příkladu jedné z platforem popsaných v této práci, konkrétně té od společnosti ZOHO. Tato platforma byla zvolena především z důvodu jednoduchosti a přehlednosti svého webového vývojového prostředí, na kterém se dá dobře ukázat, že vytvoření aplikace zvládne bez problémů i člověk, který není programátor. Dalším výběrovým faktorem byla možnost vytvoření prvních tří aplikací zcela zdarma, sice pro omezený počet uživatelů, ale pro potřeby této práce toto omezení není překážkou.

V následujících několika podkapitolách bude postupně vytvořena základní aplikace evidující prodejní objednávky k jednotlivým produktům. Cílem není vytvoření nějaké komplexní aplikace, ale spíše demonstrace stylu postupu, jakým mohou být aplikace na platformách v rámci PaaS budovány. Rozsah této práce samozřejmě neumožňuje předvést veškerou dostupnou funkcionalitu, pro hlubší prozkoumání možností platformy by čtenář měl navštívit stránky daného poskytovatele a zaměřit se na dokumentaci.

5.1 Vytvoření nové aplikace

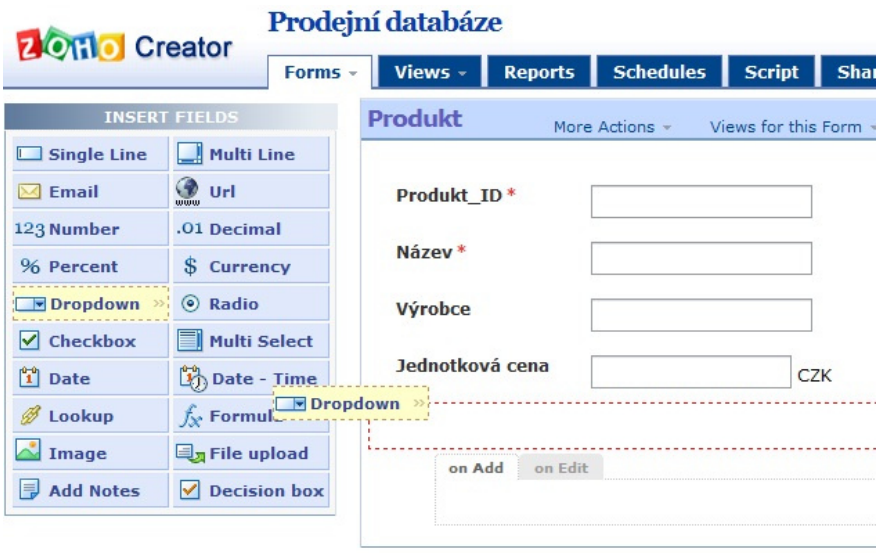
Po vytvoření uživatelského účtu a přihlášení na stránkách platformy ZOHO spustíme tzv. Creator, který nám umožní vytvořit novou aplikaci na základě předpřipraveného vzoru či zcela od začátku. My zvolíme druhou možnost a dále definujeme název aplikace a název prvního formuláře, který bude sloužit ke vkládání informací o jednotlivých produktech. Tvorbu aplikace je možno zahájit i importem dat ze souborů MS Excel či již existující databáze.



Obr. 5.1 : Vytvoření nové aplikace

5.2 Vytvoření formuláře

Pokud je aplikace vytvořena úspěšně, zobrazí se nám okno pro sestavení prvního formuláře, v našem případě s názvem Produkt. Jednotlivé řádky formuláře definujeme jednoduše přetahováním požadovaných typů polí a následným pojmenováním. K dispozici máme 20 různých typů polí. Při vytváření můžeme zároveň určovat další vlastnosti pole jako požadavek na unikátnost, povinnost vyplnění či zobrazení pole uživateli.

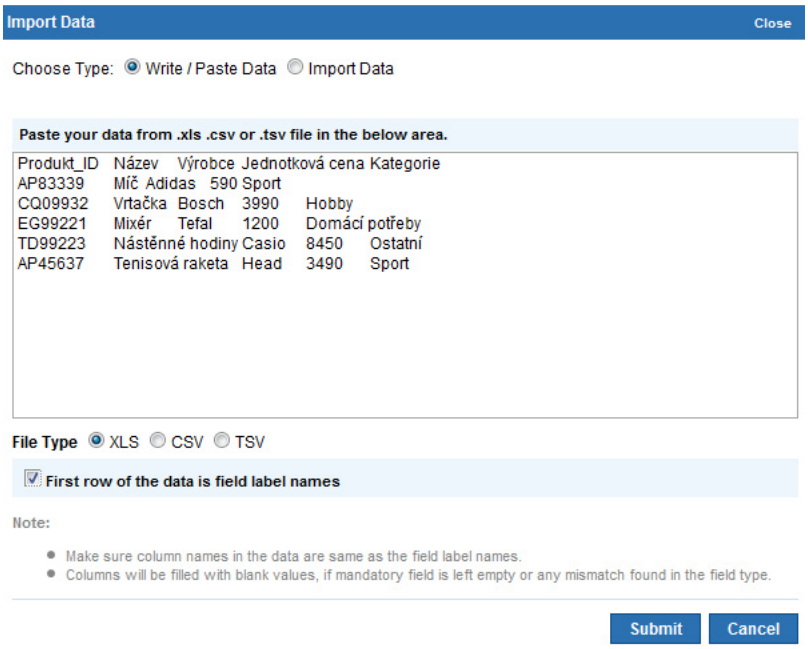


Obr. 5.2 : Sestavení formuláře z polí různého typu

Vzhled tabulky vzniklé vyplňováním dat do našeho formuláře si můžeme jednoduše zobrazit pomocí záložky Views v horní části obrazovky. Jestliže si chceme zkontrolovat, jak bude náš formulář v reálném provozu vypadat, stačí spustit naši aplikaci kliknutím na tlačítko „Access this application“.

5.3 Naplnění tabulky daty

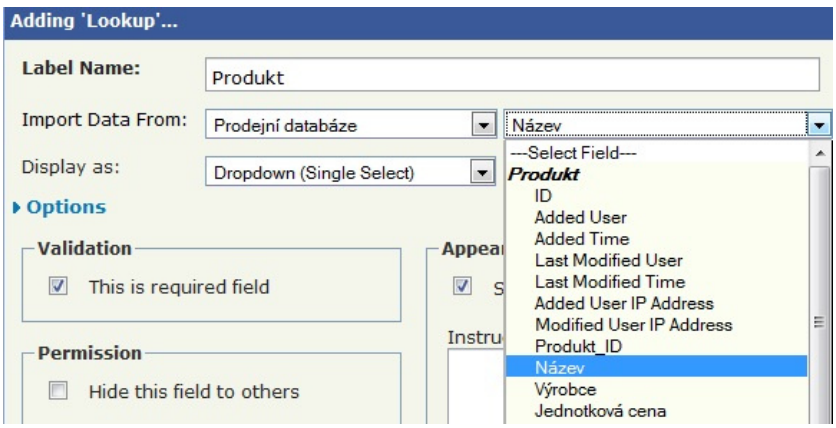
Informace o jednotlivých produktech můžeme samozřejmě do aplikace uložit postupným vyplňováním námi vytvořeného formuláře. To však může být poněkud zdlouhavé. Mnohem reálnější je situace, že tato data již máme uložena někde v lokální databázi nebo sešitu MS Excel. Nabízí se tedy možnost importu těchto dat do prostředí aplikace v ZOHO. Import můžeme provést přímo ze souboru a nebo jednoduše příkazy kopírovat/vložit, jak je zobrazeno na obrázku 5.3. Jediné, na co si musíme dát pozor, je požadavek, aby názvy sloupců importovaných dat odpovídaly názvům, které jsme zvolily při vytváření formuláře v naší aplikaci.



Obr. 5.3 : Import dat z MS Excel metodou kopírovat/vložit

5.4 Propojení formulářů

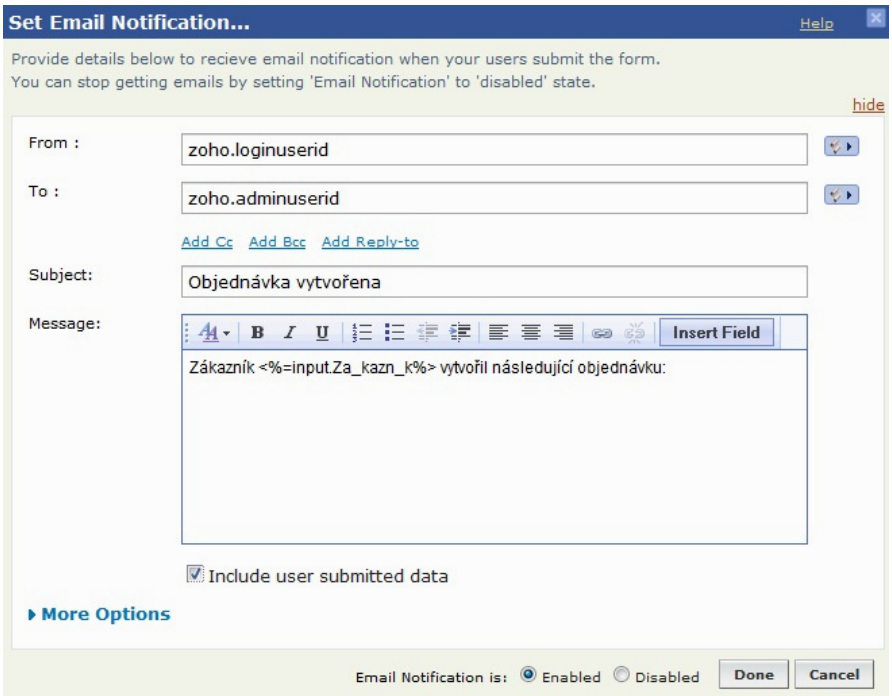
Velmi často budeme v každé naší aplikaci potřebovat, aby se data již uložená v jedné tabulce zobrazovala jako možnosti při vyplňování jiného formuláře. V našem případě budeme například chtít, aby se při vytváření nové objednávky dal ze stahovacího seznamu vybrat produkt, ke kterému se daná objednávka vztahuje. Pro jednoduchost uvažujme, že objednávka se vztahuje vždy jen k jednomu konkrétnímu produktu. Této funkcionality docílíme tak, že při vytváření formuláře Objednávka zvolíme jeden z definovaných typů pole „Lookup“, který vhodně pojmenujeme a nastavíme cestu, z které tabulky a kterého sloupce má importovat data, jak je zobrazeno na obrázku 5.4.



Obr. 5.4 : Propojení formulářů

5.5 Nastavení upozornění e-mailem

ZOHO nabízí možnost posílat upozornění na zvolený e-mail v případě, pokud dojde k vytvoření nového záznamu, smazání či modifikaci. Požadované dialogové okno spustíme ze stránky editace formuláře tlačítky **More Actions** -> **Set Email Notification**.



Obr. 5.5 : Vytvoření e-mailového upozornění

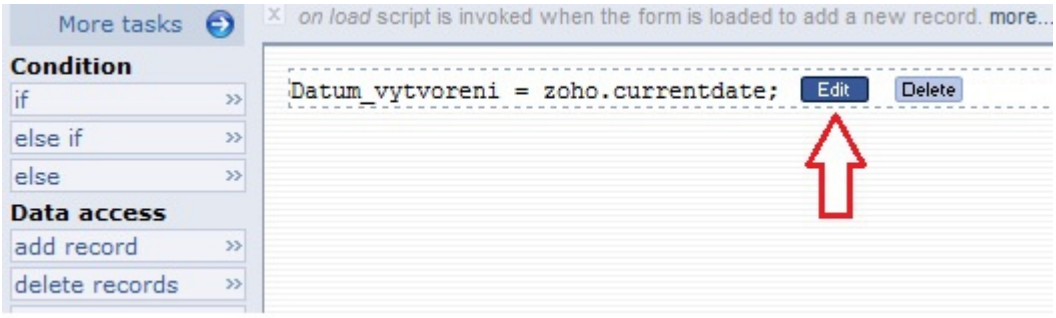
Proměnná **zoho.loginuserid** v poli From vrátí emailovou adresu právě přihlášeného uživatele, zatímco **zoho.adminuserid** vrátí email vlastníka aplikace. Vlastní tělo zprávy můžeme libovolně poskládat z různých polí aplikace, které budou načítány dynamicky. Zaškrtnutím checkboxu v dolní části dialogu zajistíme, že součástí textu budou i data, která byla do databáze právě vložena. Na závěr nesmíme ještě zapomenout nastavit emailové upozornění na hodnotu „Enabled“ a potvrdit.

5.6 Formulářové akce

Formulářové akce umožňují udělat aplikaci více dynamickou přidáním komplexní logiky do našich formulářů pomocí tzv. Deluge skriptu. Jednotlivé akce mohou být prováděny při načtení formuláře, při jeho potvrzení, při úspěšném vložení do databáze či při specifickém kliknutí.

5.6.1 Akce při načtení (On Load)

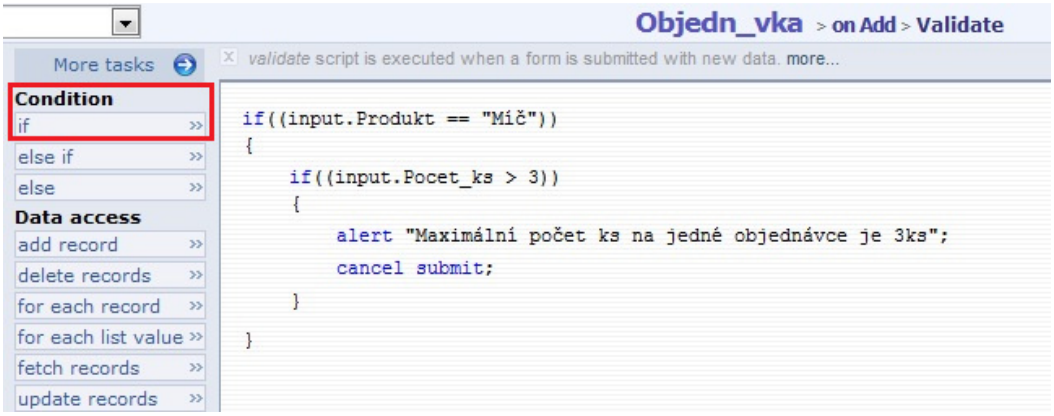
Příkladem takovéto akce může být situace, kdy máme u objednávky pole s datem vytvoření a chceme, aby se vždy při vytváření nové objednávky toto pole automaticky vyplnilo aktuálním datem. Toho dosáhneme jednoduše tak, že vybereme formulář **Objednávka** pro editaci, dále klikneme na **Form Actions** -> **on Add** -> **on Load**, přetáhneme výraz **Set Variable** do pole editoru, zvolíme **Edit**, vybereme proměnnou **Datum_vytvoreni** a jako hodnotu nastavíme proměnnou **zoho.currentdate**, která nám vrátí aktuální datum. Výsledný skript je zobrazen na obrázku 5.6.1.



Obr. 5.6.1 : Skript pro automatické nastavování data vytvoření objednávky

5.6.2 Akce při potvrzení (Validate)

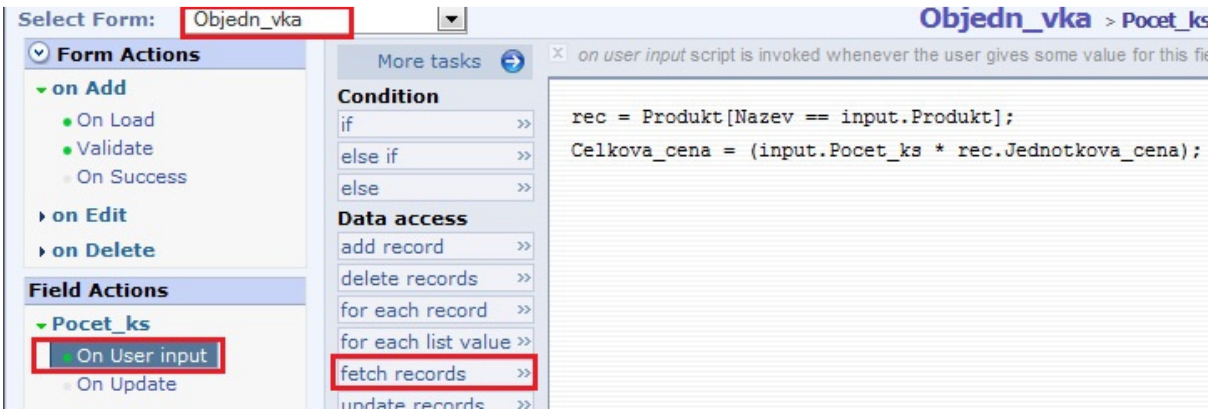
Formulář vztahující se k objednávce má i pole označující počet kusů daného produktu. Mějme modelovou situaci, že náš produkt s názvem „Míč“ je z limitované edice a je možné zakoupit na jednu objednávku maximálně 3ks. Této funkcionality docílíme takto: klikneme na **Form Actions** -> **on Add** -> **Validate**, do pole editoru přetáhneme z nabídky vlevo podmínku **If**, vybereme proměnnou **Produkt** a budeme ji porovnávat s názvem **„Míč“** (pokud budeme mít více produktů stejného názvu, bylo by patrně vhodnější použít místo názvu ID produktu). Tím zajistíme, že se naše další podmínka bude vztahovat pouze k výrobku „Míč“. Stejným způsobem vložíme do těla první podmínky druhou, tentokrát nás však zajímá proměnná **Počet_ks**. Pokud bude počet ks na objednávce větší než 3, vyskočí upozornění (**alert**) a potvrzení objednávky bude zrušeno (**cancel submit**).



Obr. 5.6.2 : Skript pro omezení počtu prodávaného zboží

5.6.3 Akce při vložení hodnoty uživatelem (On User Input)

V našem formuláři dále požadujeme, aby do pole Celková cena byla automaticky dopočítána hodnota v okamžiku, kdy uživatel zadá počet ks. On User Input je skript definovaný nad určitým polem formuláře a je vykonáván na straně klienta, to znamená dříve než, jsou data uložena do databáze. Postup bude tentokrát následující: vybereme formulář **Objednávka** a na poli **Počet ks** dáme akci **On user input**. Do editoru přetáhneme příkaz **Fetch Records** a nastavíme propojení do tabulky **Produktů** přes název produktu, jak je ukázáno na obrázku 5.6.3. Dále přetáhneme příkaz **Set Variable**, vybereme proměnnou **Celková_cena** a nastavíme výpočet, který bude součinem hodnoty v poli **Počet ks** a **Jednotkové ceny** odpovídajícího produktu. Pokaždé, když dojde ke změně v poli **Počet ks**, bude automaticky změněna i hodnota v poli **Celková cena**.

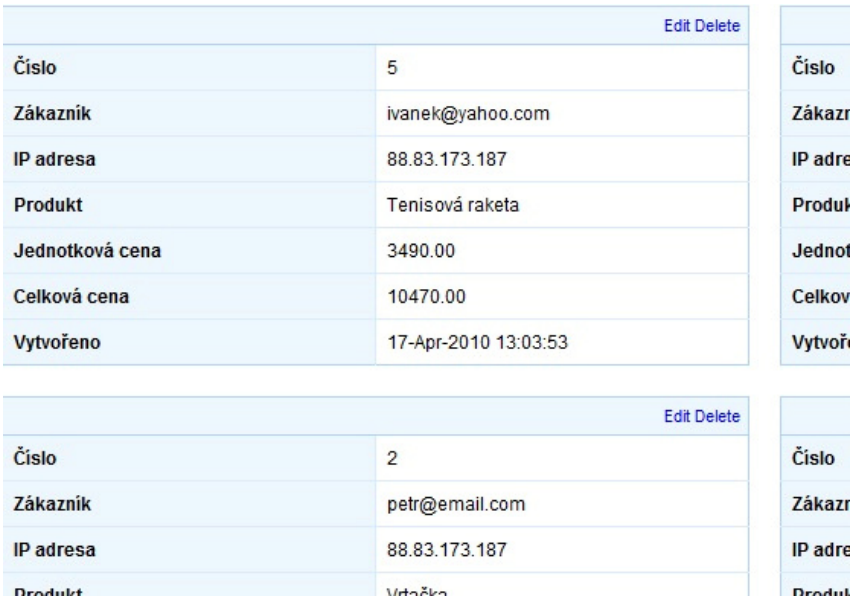


Obr. 5.6.3 : Skript pro automatický výpočet hodnoty v poli Celková cena

5.7 Pohledy (Views)

Hlavním účelem pohledů je poskytovat možnost prohlížet si data obsažená v aplikaci z různých perspektiv. Umožňují zobrazit informace z více tabulek tak, že vypadají, jako by byly umístěny v tabulce jedné. Nebo naopak můžeme vytvořit pohled, který z jedné tabulky zobrazí pouze vybrané sloupce. Základní pohledy jsou automaticky vytvářeny ke každému formuláři aplikace. Postup vytvoření vlastního pohledu se provádí jednoduše přes záložku **Views** a tlačítko **New View** v editačním módu aplikace. Kromě názvu specifikujeme i tabulku, na níž bude pohled založen, a operace, které budou moci být nad zobrazovanými daty prováděny. V následujícím dialogu pak přesně specifikujeme, co a jak má daný pohled zobrazovat. Je možné přidávat i určité systémové sloupce, jako například IP adresa uživatele, přesný čas vytvoření apod.

My v naší aplikaci vytvoříme pohled s názvem **Prehled_objednavek_view**, který bude zobrazovat přehled objednávek s následujícími poli: Číslo, Zákazník, IP adresa, Produkt, Jednotková cena, Celková cena a Vytvořeno. Zobrazujeme tedy informace z dvou různých tabulek a zároveň využíváme i dva systémové atributy. Výsledek je zobrazen na obrázku 5.7.



Obr. 5.7 : Nově vytvořený pohled na objednávky

5.8 Publikování aplikace

Po dokončení aplikace pravděpodobně budeme chtít, aby ji mohli využívat i další uživatelé. ZOHO nabízí dvě možnosti přístupu – privátní či veřejný. Veřejná aplikace je přístupná komukoliv a může být indexována webovými vyhledávači. Tato volba však není dostupná u free edice. Na druhou stranu volba privátního přístupu dává vlastníkovi aplikace možnost pozvat k užívání uživatele pomocí emailu. Ti musí mít nicméně také zaregistrovaný účet na stránkách ZOHO. Sdílena může být celá aplikace či jen její jednotlivé formuláře nebo pohledy. Publikování se nastavuje přes záložku **Share** v horní části obrazovky.

Obr. 5.8 : Pozvání k používání aplikace

5.9 Závěr

V podkapitolách 5.1 – 5.8 byly demonstrovány základní principy práce se ZOHO Creatorem. Výsledná aplikace je dostupná na webu platformy creator.zoho.com pod přihlašovacími údaji rtheier/test99. Postup práce ve webovém vývojovém prostředí byl jednoduchý, naprosto intuitivní a zvládnutelný i průměrným počítačovým uživatelem. Celý proces vývoje je neustále doprovázen různými poznámkami a radami. Pokud by přece jen někdo tápal nad určitou

funkcionalitou, k dispozici je slušně zpracovaná příručka vývojáře, ve které uživatel najde odpověď na většinu svých otázek. Pro jednoduché až středně složité databázové aplikace je tedy platforma společnosti ZOHO určitě dobrou volbou.

6 Možný budoucí vývoj

Předvídání budoucnosti je vždy velice ošidné, tím spíše pokud se jedná o tak rychle se měnící oblasti, jakými bezesporu informační technologie a ekonomie jsou. Přesto můžeme sledovat určité signály a náznaky, které nám poskytují jisté indicie, jakým směrem se asi celá oblast cloud computingu bude ubírat a jakým způsobem se bude přetvářet uvnitř.

Na úvod bych rád uvedl, že dle mého názoru je cloud computing a vůbec koncept využívání různých služeb prostřednictvím internetu velice perspektivní oblast IT. Předpokládám, že do budoucna se bude nadále vyvíjet a zvětšovat svůj podíl na trhu informačních a komunikačních technologií. V mém názoru mě utvrzuje i fakt, že své produkty spadající do této kategorie spustily i softwaroví giganti jako Microsoft či Google a další velcí hráči jako IBM nebo Oracle již postupně také přichází se svými řešeními. To by mohlo zvýšit důvěru zákazníků v celou oblast, protože asi málokdo si dokáže v blízké budoucnosti představit krach některé z těchto společností.

Rozšiřování cloud computingu může paradoxně nahrávat i současná ekonomická krize, která vypukla v roce 2008. Většina společností byla nucena v rámci úsporných opatření seškrtat své výdaje, což se samozřejmě nevyhnulo ani IT oddělením. Kvůli nižším disponibilním rozpočtům nastal čas hledání levnějších alternativ ke klasickému provozování podnikové informatiky. A takovou alternativou může být právě cloud computing, který přináší možnost ušetřit peníze za rozsáhlé investice do infrastruktury a i do její následné obsluhy. Pomyslný míč je teď na straně poskytovatelů služeb. Ti musí přesvědčit své potencionální zákazníky o bezpečnosti a spolehlivosti nabízených řešení a zároveň tyto skutečnosti dostatečným způsobem garantovat. Podniky totiž budou porovnávat případné ekonomické přínosy s bezpečnostními riziky, která jsou často v souvislosti s cloud computingem zmiňována. S tím částečně souvisí i nutnost určení a zavedení určitých standardů do celé oblasti, které by umožňovaly změnu poskytovatele s minimálními peněžními a časovými náklady. Jejich současná neexistence může řadu potencionálních klientů odradit ze strachu z „uzamčení“ jejich dat a aplikací u jednoho konkrétního poskytovatele. To je rizikové z toho hlediska, že daný poskytovatel může znenadání

zvýšit ceny svých služeb či dokonce ukončit činnost, což by pro klienta znamenalo nutnost větších či menších úprav aplikace tak, aby mohla být provozována v jiném prostředí.

Dá se předpokládat, že služby spadající do cloud computingu budou rychleji přijímány a využívány začínajícími a menšími společnostmi, opět z důvodu minimálních počátečních nákladů a rychlosti implementace. Pro tento druh společností mohou například SaaS aplikace představovat nejenom úsporu na straně nákladů, ale i bonus co se týče samotné funkcionality a s tím související zlepšení podnikových procesů. Velké společnosti budou patrně při adopci poněkud zdrženlivější. Za nemalé peníze mají vybudovanou rozsáhlou interní infrastrukturu, která pravděpodobně bude svou funkcionalitou pro aktuální potřeby dostačující a není tedy nutné ze dne na den něco měnit. I oni však jistě budou boom okolo cloud computingu bedlivě sledovat a začnou jeho služeb využívat alespoň na menší projekty v rámci jednotlivých oddělení nebo i na větší aplikace, které ale nemají kritický význam pro chod společnosti. Čeká nás tedy doba kombinování klasicky dodávaného softwaru a služeb cloud computingu v závislosti na typu a charakteru dané aplikace. Více než public cloudy budou preferovány soukromé private cloudy umístěné za podnikovým firewallem. Společnost tak bude mít větší pocit bezpečí svých dat a zároveň jí to i do budoucna usnadní případný přechod do veřejného cloudu. Naopak u velkých finančních institucí, jakými jsou banky nebo pojišťovny, nemůžeme asi v blízké budoucnosti očekávat nějaký enormní zájem o oblast cloud computingu. Společnosti tohoto typu dbají v maximální možné míře na bezpečnost dat a využívají jen detailně prověřené technologie, což cloud computing i z důvodu své poměrně krátké existence samozřejmě prozatím nesplňuje.

Co se týká samotného modelu Platform-as-a-Service, tak i zde vidím velký prostor pro další rozvoj a rozšiřování. Jedním z hlavních přínosů je dle mého názoru usnadnění tvorby webových aplikací. V současnosti, pokud chce někdo vytvořit nějakou takovou aplikaci, musí znát celou řadu různých technologií jako PHP, MySQL, JavaScript, Apache, http a mnohé další. PaaS toto v mnohém směru velmi ulehčuje a dává vývojáři možnost soustředit se na to hlavní, což je obsah a funkcionalita aplikace. Je velice pravděpodobné, že i na trhu PaaS budou do budoucna hrát hlavní roli platformy velkých společností jako Google nebo Microsoft, které sice svá řešení spustily později, ale o to více je nyní budou propagovat a „nutit“ uživatele prostřednictvím svých dalších produktů využívat služeb jejich cloudu.

Jaký bude skutečný vývoj celé oblasti cloud computingu, to ukáže až čas. Může skončit pouze na stránkách odborné literatury jako další slibná technologie, která se však neuchytila, a nebo

naopak za několik let bude hrát dominantní úlohu na celém poli informačních technologií. Nechme se tedy překvapit, co budoucnost přinese.

7 Závěr

Pojem cloud computing a celá oblast s ním související je v současnosti předmětem velkého zájmu odborné i laické veřejnosti. Nejedná se o žádnou novou převratnou technologii, ale spíše o změnu uvažování o přístupu k informačním technologiím a využívání stávajících technologií novými způsoby. I samotná myšlenka IT jako služby není nová, ale nyní se zdá, že konečně nastal její čas.

Tato práce se ve své první části zaměřila na vymezení cloud computingu jako takového včetně základních pojmů s tímto tématem souvisejících. Celá oblast nemá dosud jednoznačně určenou terminologii, ale postupně dochází k přijímání a postupnému vžívání anglických termínů. Dále byly popsány jednotlivé aspekty, které by měly uživatele k využívání služeb v rámci cloud computingu motivovat, z nichž nejvýznamnější úlohu hrají ty ekonomické.

Druhá a třetí část práce se podrobněji věnovaly jednomu ze základních modelů cloud computingu Platform-as-a-Service. Kromě základního popisu byla i na příkladu společnosti Force.com detailněji popsána architektura takové platformy. Zároveň byla provedena analýza trhu s produkty PaaS za účelem zjištění, jaká veškerá funkcionalita je v rámci této oblasti zákazníkům v současnosti dostupná. V průběhu zkoumání trhu došlo k poněkud překvapivému zjištění, že platforem, které svým zaměřením spadají do kategorie PaaS, je ve skutečnosti více, než ukazovaly původní předpoklady na začátku zpracovávání tohoto tématu. K detailnějšímu popisu bylo nakonec vybráno sedmnáct platforem, které byly rozděleny do tří kategorií. Zároveň bylo zjištěno, že spektrum nabízených řešení je opravdu široké a některé platformy se od ostatních výrazně liší. Do budoucna bude tedy s přibývajícím počtem dalších hráčů na trhu pravděpodobně nezbytné celou oblast PaaS podrobněji kategorizovat podle účelu, pro jaký jsou jednotlivé platformy primárně určeny. Dalším zjištěním je fakt, že v současnosti neexistuje v rámci PaaS platforma, která by poskytovala webové prostředí pro online vývoj aplikací založené na některém standardním programovacím jazyku.

Ve čtvrté části bylo demonstrováno použití jedné z platforem na příkladu tvorby jednoduché databázové aplikace na evidenci objednávek. Cílem bylo ukázat proces vývoje takové aplikace online přímo v prostředí internetového prohlížeče za pomoci drag and drop nástrojů i vytváření jednoduchých skriptů.

V samotném závěru práce je naznačen možný vývoj nejbližší budoucnosti cloud computingu tak, jak ho vidí autor této práce.

8 Seznam použitých zdrojů

[AMA2009] Amazon Web Services: *Overview of Security Processes*, November 2009 [online] URL:< http://awsmedia.s3.amazonaws.com/pdf/AWS_Security_Whitepaper.pdf> [cit 2010-03-06]

[ARM2009] ARMBRUST, M. a kol.: Electrical Engineering and Computer Sciences, University of California at Berkeley, 2009. *Above the Clouds: A Berkeley View of Cloud Computing* : *Technical Report No. UCB/EECS-2009-28*. [online] URL:<http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf> [cit. 2010-02-22]

[BAB2009] BABCOCK, Ch. InformationWeek, *Platform As A Service: What Vendors Offer*, October 2009 [online] URL:<http://www.informationweek.com/news/services/saas/showArticle.jhtml;jsessionid=ADCID5T54KGK5QE1GHPSKHWATMY32JVN?articleID=220300686&pgno=2&queryText=&isPrev=> [cit 2010-04-08]

[BRU1998] BRUCKNER, T. - VOŘÍŠEK, J.: *Outsourcing informačních systémů*. Praha, EKOPRESS, 1998, ISBN 80-86119-07-6

[BUR2008] BURKOŇ, L. *Software as a Service 2.0*. Praha: Vysoká škola ekonomická, Katedra informačních technologií, 2008. 88 s. Vedoucí diplomové práce: Dr. George Feuerlicht.

[CAR2003] CARR, N. *IT Doesn't Matter*. *Harvard Business Review*, 2003 [online] URL:<http://www.econ.uba.ar/www/departamentos/sistemas/plan97/tecn_informac/rota/Zimerman/IT%20Doesnt%20matter%20Carr.pdf> [cit. 2010-02-18]

[DAT2010] STANKOV, I., DATSENKA, R. *Platform-as-a-Service as an Enabler for Global Software Development and Delivery* [online] URL:<http://webdoc.sub.gwdg.de/univerlag/2010/mkwi/01_management_und_methoden/software-industrie/09_platform-as-a-service_as_an_%20enabler_for_global_software_development_and_delivery.pdf > [cit 2010-03-11]

[DNA2010] DNA India, *Bangalore´s cloud cover*, January 2010 [online] URL:< http://www.dnaindia.com/bangalore/report_bangalore-s-cloud-cover_1336482> [cit 2010-04-04]

[DOC2010] Developer force: *Documentation* [online] URL:< http://wiki.developerforce.com/index.php/Documentation> [cit 2010-03-23]

[GAL2006] GÁLA, L., POUR, J., TOMAN, P. *Podniková informatika.* 1. vyd. Praha: Grada Publishing, 2006. 482 s. ISBN 80-247-1278-4

[GAL2009] GÁLA, L., POUR, J., ŠEDIVÁ, Z. *Podniková informatika*, 2. vyd. Praha: Grada Publishing, 2009, 496 s. ISBN 80-247-2615-1

[GAR2009] GARTNER: *Gartner´s 2009 Hype Cycle Special Report* [online] URL: http://www.gartner.com/it/page.jsp?id=1124212 [cit. 2010-02-18]

[HAJ2007] HÁJEK, J. *Virtualizace – mýtus, kouzlo, hype nebo realita?*, 2007, [online] URL:< http://interval.cz/clanky/virtualizace-mytus-kouzlo-hype-nebo-realita/> [cit 2010-02-27]

[HIN2008] HINCLIFFE, D. *The Next Evolution in Web Apps: Platform-as-a-Service (PaaS)* [online]
URL:< http://blogs.bungeeconnect.com/2008/04/28/free-white-paper-the-next-evolution-in-web-apps-platform-as-a-service-paas/> [cit 2010-03-10]

[HOL2009] HOLAS, P. *Google App Engine* [video online]
URL:< http://www.youtube.com/watch?v=srsyNqpMRp8> [cit 2010-04-08]

[CHA2009] CHAPPELL, D. *Introducing The Windows Azure Platform, v1.3*, December 2009 [online]
URL:<http://view.atdmt.com/action/mrtyou_FY10AzurewhitepapterIntroWindowsAzurePl_1> [cit 2010-04-07]

[IDC2009] PERRY, R. a kol., Force.*com Cloud Platform Drives Huge Time to Market and Cost Savings*, September 2009 [online]
URL:<http://thecloud.appirio.com/rs/appirio/images/IDC_Force.com_ROI_Study.pdf > [cit 2010-03-16]

[JOH2008] JOHNSTON, S. Taxonomy: *The 6 layer Cloud Computing Stack* [online]
URL:< http://samj.net/2008/09/taxonomy-6-layer-cloud-computing-stack.html> [cit 2010-03-04]

[KOD2008] KODERA, J. *Jak vypadají oblaka* [online]
URL:< http://digitalne.centrum.cz/jak-vypadaji-oblaka/> [cit 2010-02-26]

[LAW2008] LAWTON, G. *Developing Software Online With Platform-as-a-Service Technology*, Computer, vol. 41, no. 6, pp. 13-15, June 2008 [online]
URL:< http://www.computer.org/portal/web/csdl/doi/10.1109/MC.2008.185 > [cit 2010-03-11]

[MAL2009] MALÝ, M. *REST: architektura pro webové API*, Srpen 2009, [online]
URL:< http://zdrojak.root.cz/clanky/rest-architektura-pro-webove-api/> [cit 2010-04-27]

[MEL2009] MELL, P., GRANCE, T.: National Institute of Standards and Technology , 2009. *The NIST Definition of Cloud Computing.* [online] URL:<http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc > [cit. 2010-02-18]

[MIL2008] MILLER, R. *Google Data Cente FAQ* [online] URL:< http://www.datacenterknowledge.com/archives/2008/03/27/google-data-center-faq/> [cit 2010-02-27]

[MIT2008] MITCHELL, D. *Defining Platform-As-A-Service, or PaaS*, February 2008 [online] URL:< http://blogs.bungeeconnect.com/2008/02/18/defining-platform-as-a-service-or-paas/> [cit 2010-03-11]

[MOO2009] MOORE, D. InformationWeek Analytics: *Cloud Computing: Platform As A Service – Weigh The Risks And Benefits Before Deciding*, October 2009 [online] URL:< http://analytics.informationweek.com/abstract/5/1350/Cloud-Computing/cloud-computing-platform-as-a-service.html> [cit 2010-03-24]

[NAT2009] NATIS, Y. a kol. *Who´s Who in Application Platforms for Cloud Computing: The Cloud Specialists*, September 2009 [online] URL:< http://www.gartner.com/DisplayDocument?id=1180217> [cit 2010-04-04]

[NET2009] NetSuite, *NetSuite Offers Green Alternative to Traditional On-Premise ERP and CRM Solutions*, July 2009 [online] URL:< www.netsuite.com/portal/pdf/netsuite-green-wp.pdf> [cit 2010-04-20]

[NUC2009] NUCLEUS RESEARCH, *Force.com drives faster development*, May 2009 [online] URL:< http://nucleusresearch.com/research/notes-and-reports/force-dot-com-drives-faster-development/ > [cit 2010-03-16]

[OPM2009] *Open Cloud Manifesto*, 2009 [online] URL:< http://www.opencloudmanifesto.org/opencloudmanifesto1.htm> [cit 2010-03-07]

[PIE2009] PIECH, M. An Oracle White Paper: Platform-*as-a-Service Private Cloud with Oracle Fusion Middleware* [online] URL:<http://www.oracle.com/ocom/groups/public/documents/webcontent/036500.pdf> [cit 2010-03-04]

[PLU2009] PLUMMER, D. Gartner: *Experts Define Cloud Computing: Can we get a Little Definition in our definitions?* [online] URL:< http://blogs.gartner.com/daryl_plummer/2009/01/27/experts-define-cloud-computing-can-we-get-a-little-definition-in-our-definitions/> [cit. 2010-02-18]

[REE2009] REESE, G. *Cloud Application Architectures*. Sebastopol, CA (USA) : O'Reilly Media, Inc., 2009. 189 s. ISBN : 978-0-596-15636-7

[RYM2009] RYMER, J., *Platform-As-A-Service Is Here; Can It Help You?*, February 2009 [online] URL:< http://www.microfocus.com/000/Forrester-Platform-As-A-Service%20Is%20Here_tcm21-25854.pdf > [cit 2010-04-22]

[SAL2007] Apex Code: *The World´s First On-Demand Programming Language* [online] URL:<http://wiki.developerforce.com/index.php/Apex_Code:_The_World%27s_First_On-Demand_Programming_Language> [cit 2010-03-23]

[SAL2010] The world´s first real-time cloud [online] URL:< http://www.salesforce.com/eu/platform/cloud-infrastructure/> [cit 2010-03-16]

[STA2008] JOHNSON, B. 2008. *Cloud computing is a trap, warns GNU founder Richard Stallman*. [online] URL:<http://www.guardian.co.uk/technology/2008/sep/29/cloud.computing.richard.stallman> [cit. 2010-02-18]

[TVA2009] TVARŮŽEK, S. *The Shift Towards Cloud Computing*, Master Thesis, Masaryk Universtity, Brno, May 2009 [online] URL:< http://is.muni.cz/th/51532/fi_m/ > [cit 2010-03-05]

[VOR2008] VOŘÍŠEK, J. a kol. *Principy a modely řízení podnikové informatiky*. 1. vyd. Praha: Oeconomica, 2008. 446 s. ISBN 978-80-245-1440-6

[W3C2010] W3C: *Scripting and AJAX* [online] URL:< http://www.w3.org/standards/webdesign/script.html> [cit 2010-04-27]

[WEI2009] WEISSMAN, C., BOBROWSKI, S. *The Design of the Force.com Multitenant Internet Application Development Platform* [online] URL:<http://portal.acm.org.ezproxy.vse.cz/ft_gateway.cfm?id=1559942&type=pdf&coll=ACM&dl=ACM&CFID=81661987&CFTOKEN=10496736 > [cit 2010-03-12]

9 Terminologický slovník

Termín	**Zkratka**	**Význam**
Asynchronous JavaScript and XML	AJAX	Skupina technik, které umožňují dynamicky měnit obsah webové stránky bez nutnosti jejího kompletního znovunačtení. [W3C2010]
Application Programming Interface	API	Aplikační programové rozhraní – množina pravidel, která upravuje komunikaci mezi aplikacemi, nebo jednotlivými částmi aplikací. [GAL2006, str.223]
Buzzword		Módní pojem, který velmi vágně vymezuje určitou problematiku, často bývá marketingového původu. [autor]
Customer Relationship Management	CRM	Řízení vztahů se zákazníky [autor]
Central Processing Unit	CPU	Procesor počítače [autor]

Drag and drop		Způsob zpřístupnění akcí a manipulace s virtuálními objekty pomocí myši. [autor]
Enterprise Resource Planning	ERP	Typ aplikace, resp. aplikačního software, který umožňuje řízení a koordinaci všech disponibilních podnikových zdrojů a aktivit. [GAL2009, str.160]
Integrated Development Environment	IDE	Integrované vývojové prostředí. Software poskytující programátorům nejrůznější nástroje pro usnadnění vývoje aplikací. [autor]
Lock-in		Označení pro situaci, kdy data či aplikace nemůže být jednoduše přenesena k jinému poskytovateli. [autor]
Mash-up		Webová aplikace spojující data a funkcionalitu z více zdrojů a tvořící tak novou službu. [autor]
Metadata		Data popisující jiná data. Jejich prostřednictvím se lze na data dotázat, jsme schopni data doplňovat, konsolidovat je, vzájemně je synchronizovat a integrovat. [GAL2009, str.62]
Multi-tenancy		Webová aplikační architektura umožňující chod pouze jedné instance aplikace společné pro všechny zákazníky využívající tuto aplikaci. [BUR2008, str.84]
Outsourcing		Zajišťování vybraných činností a služeb externími dodavateli. [GAL2009, str.383]
Pay-as-you-go	PAYGO	Způsob účtování služeb, kdy zákazník platí pouze za to, co skutečně využívá (neplatí žádné paušální či licenční poplatky). [autor]
Representational State Transfer	REST	Architektura rozhraní, které je použitelné pro jednotný a snadný přístup ke zdrojům. [MAL2009]
Search Engine Optimization	SEO	Metody pro tvorbu a úpravu webových stránek, které mají za cíl co nejvyšší postavení daných stránek ve výsledcích internetových vyhledávačů. [autor]
Simple Object Access Protocol	SOAP	Protokol umožňující přenášet XML dokumenty. [GAL2009, str.491]
Software Development Kit	SDK	Nástroje pro podporu snadnějšího vývoje aplikací spolupracující s vybraným vývojovým prostředím. [GAL2009, str.49]

10 Přílohy

10.1 Příloha A – Tabulka poskytovatelů z kapitoly 4.1

	Základní ceny	Způsob účtování	"Tržiště" s již hotovými aplikacemi	Možnost vývoje offline
ZOHO	zdarma - 5$/uživatel/měsíc	počet uživatelů, záznamů	ano	ne
Caspio	39,95$ - 349,95$/měsíc	počet stránek, velikost úložiště, počet typů uživatelů	ne	ne
DabbleDB	zdarma - 8$/měsíc	soukromá/veřejná data, počet uživatelů	ne	ne
Teamdesk	49$ - 249$/měsíc	počet aplikací, počet uživatelů	ano	ne
Qrimp	2$ - 25$/uživatel/měsíc	počet uživatelů, velikost úložiště	ne	ano
Wavemaker	v závislosti na poskytovateli IaaS	v závislosti na poskytovateli IaaS	ne	ano
WOLF	zdarma - 10$/uživatel/měsíc	počet uživatelů, velikost úložiště	ano	ne
LongJump	30$ - 90$ /měsíc	počet objektů, velikost úložiště pro 1 uživatele	ano	ano
OrangeScape	nejasné, zřejmě 15$/uživatel	nejasné	ano	ne
WorkXpress	25,5$ - 30$	dle poskytovatele IaaS	ano	ne
Bungee Connect	zatím nezveřejněno	zatím nezveřejněno	ne	ne
Force.com	zdarma - 75$/uživatel/měsíc	počet uživatelů, počet db objektů	ano	ano
Iceberg	10$/uživatel/měsíc	počet uživatelů (možnost individ. cen)	ne	ne
Rollbase	49$/uživatel	počet záznamů a portálů, velikost úložiště	ano	ne

Tabulka 10.1 - Přehled poskytovatelů platforem pro vývoj aplikací v prohlížeči

10.2 Příloha B – Tabulka poskytovatelů z kapitoly 4.2

	Programovací jazyky	Vývojové prostředí	Způsob účtování
MS Azure	jakýkoliv .NET jazyk	Visual Studio 2008/2010	pay as you go nebo závazek na delší dobu, ceník podle 3 základních komponent
Google App Engine	Python, Java	Eclipse	do určitých limitů zdarma, poté podle jednotlivých zdrojů
Stax	Java a další jazyky využívající JVM	Eclipse nebo jiné IDE	zatím nezveřejněno
Heroku	Ruby	libovolný textový editor, jakékoliv IDE podporující Ruby	možnost sestavení měsíčního účtu dle individuálních požadavků na zdroje
Engine Yard	Ruby	libovolný textový editor, jakékoliv IDE podporující Ruby	hodinová sazba podle velikosti instance

Tabulka 10.2 – Přehled poskytovatelů platforem pro lokální vývoj aplikací

10.3 Příloha C – Seznam obrázků použitých v textu

Obrázek 2.9 : Různé modely cloud computingu

Obrázek 3.2.1 : Ukázka multitenant aplikace, která obsluhuje několik uživatelů

Obrázek 3.2.2 : Schéma architektury řízené metadaty

Obrázek 3.2.3 : Model ukládání dat u Force.com

Obrázek 3.3.1 :Celkové náklady na aplikaci

Obrázek 4.2.1: Schéma fungování platformy Azure

Obrázek 5.1 : Vytvoření nové aplikace

Obrázek 5.2 : Sestavení formuláře z polí různého typu

Obrázek 5.3 : Import dat z MS Excel metodou kopírovat/vložit

Obrázek 5.4 : Propojení formulářů

Obrázek 5.5 : Vytvoření e-mailového upozornění

Obrázek 5.6.1 : Skript pro automatické nastavování data vytvoření objednávky

Obrázek 5.6.2 : Skript pro omezení počtu prodávaného zboží

Obrázek 5.6.3 : Skript pro automatický výpočet hodnoty v poli Celková cena

Obrázek 5.7 : Nově vytvořený pohled na objednávky

Obrázek 5.8 : Pozvání k používání aplikace