

Srovnání modelů dodávky a provozu aplikací

verze 1.0, 20/11/2024

Obsah

1.	Účel dokumentu	2
2.	Srovnání modelů dodávky a provozu aplikací graficky	3
3.	Stručné srovnání modelů	4
4.	Detailní srovnání modelů	6
5.	Význam použitých zkratk	20
6.	Použité zdroje	21

1. Účel dokumentu

Tento dokument je určen jako podpůrný metodický materiál manažerům OVS, kteří zodpovídají za výběr provozního modelu využívaných ISVS.

Na modely dodávky a provozu aplikací organizace lze nahlížet z mnoha různých pohledů, např.:

- a) Jde o aplikaci, jejíž funkcionalita je „šitá“ na míru potřebám dané organizace, nebo jde o standardizovanou aplikaci (jako je ERP, HR, spisová služba, e-mail apod.), kterou výrobce dodává ve standardizované podobě všem zákazníkům?
- b) Je aplikace multi-tenantní nebo single-tenantní?
- c) Kdo zodpovídá za provoz a podporu aplikace – poskytovatel nebo zákazník?

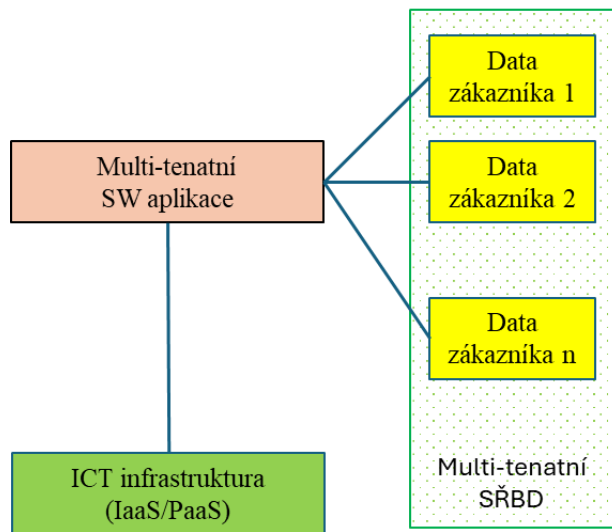
Každý z modelů dodávky a provozu aplikace má své výhody i nevýhody, které musí organizace ve svých konkrétních podmínkách před volbou modelu posoudit a následně kvalifikovaně rozhodnout.

Dokument porovnává tři základní modely provozu aplikací. Řešení prvních dvou modelů je ovlivněno regulací cloud computingu vyplývající ze zákona č. 365/2000 Sb., o informačních systémech veřejné správy (ZoSVS). Dopady této regulace jsou detailně popsány v samostatném dokumentu „Požadavky na zápis aplikací jako služba v nuancích provozních modelů“¹.

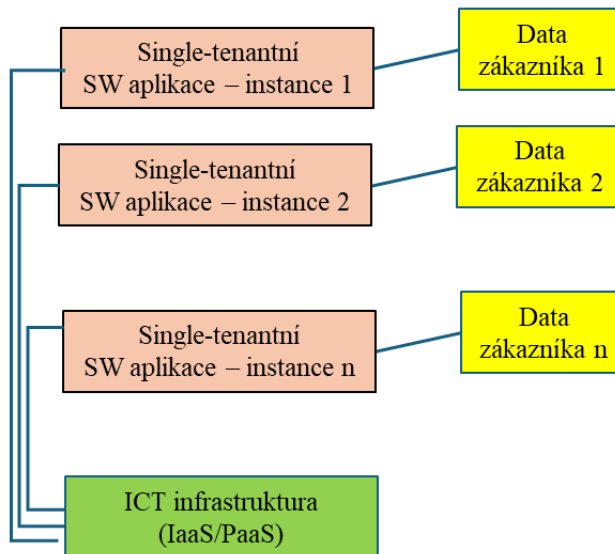
¹ Dostupné zde [Podpůrné materiály – Digitální a informační agentura](#)

2. Srovnání modelů dodávky a provozu aplikací graficky

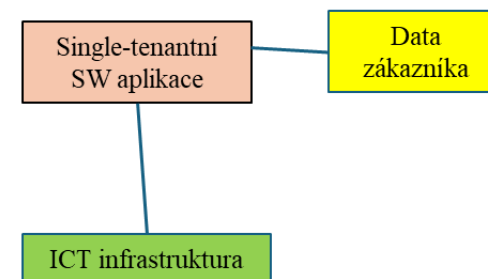
Multi-tenantní aplikace (SaaS)



**Single-tenantní aplikace
na IaaS/PaaS poskytovatele**



**Single-tenantní aplikace
na ICT infrastruktuře zákazníka**



3. Stručné srovnání modelů

- **Multi-tenantní aplikace provozovaná na IaaS/PaaS:** Jedná se o sdílené prostředí, kde více zákazníků využívá současně jednu instanci aplikace.
- **Single-tenantní aplikace provozovaná na IaaS/PaaS:** Každý zákazník má vlastní instanci aplikace běžící na sdíleném prostředí IaaS/PaaS.
- **Single-tenantní aplikace v on-premise provozu:** Každý zákazník má vlastní instalaci aplikace na svém technologickém prostředí.

Srovnání dle nejpodstatnějších kritérií:

Kritérium / Model aplikace	Multi-tenantní nad IaaS/PaaS	Single-tenantní nad IaaS/PaaS	Single-tenantní (on-premise)
Náklady	Nižší počáteční, variabilní	Počáteční nižší než on-premise, variabilní. Vyšší náklady než při multi-tenantním provozu	Vysoké počáteční i fixní
Škálovatelnost	Vysoká, automatická	Vysoká, automatická	Nízká, manuální
Dostupnost	Vysoká, díky redundanci výpočetních zdrojů	Vysoká, díky redundanci výpočetních zdrojů	Závisí na infrastruktuře
Bezpečnost	Může být nižší, závisí na poskytovateli	Vyšší než multi-tenantní	Může být vysoká, je pod plnou kontrolou zákazníka
Přizpůsobení	Omezené	Vysoké	Vysoké
Požadavky na správu a údržbu zákazníkem	Minimální	Nižší než on-premise	Vysoké
Rychlost nasazení	Rychlé	Rychlé	Pomalejší
Flexibilita	Vysoká	Vysoká	Nízká
Pro koho vhodné	Pro malé a střední organizace, které nepotřebují vysokou míru přizpůsobení funkcionality aplikace.	Pro organizace, které potřebují vysokou míru přizpůsobení a zároveň chtějí využít výhod cloudu.	Pro velké organizace s vysokými nároky na bezpečnost a výkon. Pro aplikace vyžadující specifické konfigurace.

Kromě výše uvedených kritérií je vhodné zvážit také následující aspekty:

- Licencování: Náklady na software a jeho licencování se mohou lišit v závislosti na modelu provozu.
- Zálohování a obnovení: Různé modely nabízejí různé možnosti zálohování a obnovy dat.
- Shoda s legislativou: Některé aplikace musí splňovat specifické regulační požadavky, které mohou ovlivnit volbu modelu.
- Integrovaní možnosti: Schopnost integrace s dalšími systémy a aplikacemi může být důležitým faktorem.
- Podpora: Dostupnost a kvalita podpory poskytovatele může ovlivnit celkovou spokojenost zákazníka.

Doporučení:

Pro výběr nejvhodnějšího modelu provozu je důležité provést důkladnou analýzu konkrétních požadavků a omezení projektu.

4. Detailní srovnání modelů

Poznámka: Popis rozdílů mezi třemi formami realizace softwarové aplikace vychází z v praxi převažujících situací. V konkrétních podmínkách může docházet k odchylkám. Tabulka ale může pomoci zákazníkovi uvědomit si, jaký vliv má odchylka na flexibilitu, bezpečnost a nákladovost jeho řešení.

Charakteristika	SaaS model (čistě multi-tenantní model)	Single-tenantní aplikace na cloudové platformě IaaS/PaaS	On-premise (Single-tenantní aplikace na ICT infrastruktuře zákazníka)
Základní princip	Materiální poskytovatel udržuje, rozvíjí, zajišťuje provoz a dává k dispozici multi-tenantní aplikaci, multi-tenantní systém řízení báze dat a multi-tenantní ICT infrastrukturu pro provoz aplikace a podpůrné služby s aplikací související; dodává je velkému počtu zákazníků prostřednictvím internetu jako službu. Mnoho uživatelů z různých organizací využívá tutéž instanci aplikace společně. [1][2][3][4][5][6] Konverze single-tenantní aplikace na multi-tenantní je náročný proces. Pro menší SW firmy konverze může vyžadovat i několik let vývoje a testování.	Pro účely tohoto srovnání jsou v této skupině zahrnuty smíšené modely, které kombinují určité single-tenantní komponenty s jinými multi-tenantními komponentami, tak jak to může být pro poskytovatele výhodné z důvodu minimalizace nákladů. [7] Výrobce aplikačního SW (který může a nemusí být v roli materiálního poskytovatele) vytvoří a dále rozvíjí single-tenantní aplikaci, která je provozována na cloudové platformě IaaS/PaaS. Tato cloudová platforma IaaS/PaaS může být zajištěna: (i) třetí stranou, odlišnou od poskytovatele aplikace, který zde vystupuje v roli přeprodejce IaaS/PaaS (ii) na vlastní cloudové platformě IaaS/PaaS poskytovatele aplikace (iii) v tenantu IaaS/PaaS zřízeném zákazníkem (OVS).	Výrobce SW vytvoří a dále rozvíjí single-tenantní aplikaci, která je instalována v datovém centru zákazníka obvykle specializovanou firmou. Za provoz ICT infrastruktury, za instalaci nových verzí SW a za dodávku služeb uživatelům je zodpovědný interní ICT útvar zákazníka. Tuto svoji zodpovědnost však může částečně přenést na specializovanou firmu.

		Za provoz ICT infrastruktury, za instalaci nových verzí SW a za dodávku služeb uživatelům je zodpovědný poskytovatel (pokud si zákazník sám neadministruje aplikační SW v jím zřízeném tenantu IaaS/PaaS – viz bod (iii) výše).	
--	--	---	--

Technologické aspekty			
Východisko návrhu aplikace	Od počátku navrhována pro provoz specializovanou firmou a pro dodávku služby přes internet velkému počtu zákazníků, tzn. že specificky navrženy jsou: HW a SW architektura a na ně navazující obchodní model a procesní model (uzavírání smluv, dodávka a účtování služeb, kontrola plnění SLA, ...). Využívá se zejména pro standardizované aplikace jako jsou ERP, HR, CRM, spisová služba, e-mail, u kterých se určitý segment zákazníků (typicky malé a střední organizace) spokojí s malým rozsahem customizace. Odlišné fungování aplikace pro různé zákazníky jde totiž proti základní logice modelu – využívat jednu instanci kódu pro všechny zákazníky současně. Customizace se neprovádí změnou kódu, ale hodnotami parametrů, které ovlivňují funkčnost aplikace pro daného zákazníka.	Návrh přizpůsoben té skutečnosti, že poskytovatel instaluje na infrastrukturu IaaS/PaaS pro každého zákazníka samostatnou instanci SW. Její přizpůsobení požadavkům zákazníka probíhá podobně jako u on-premise.	Návrh aplikace a dělba práce při instalaci a provozu aplikace jsou přizpůsobeny té skutečnosti, že instalaci a přizpůsobení aplikace technologickému prostředí a byznysovým požadavkům zákazníka může provádět specializovaná firma (často jiná než výrobce). Tato firma při instalaci využívá customizační parametry, metodiky a nástroje implementace, nástroje pro správu a distribuci nových verzí aplikace. Při instalaci firma využívá konzultanty specializované na customizaci jednotlivých modulů aplikace (např. výroba, obchod, účetnictví apod.). Přizpůsobení aplikace požadavkům zákazníka se realizuje nejen customizačními parametry, ale i změnou kódu aplikace. Zákazník je zodpovědný za správu a provoz aplikace, ale může ji částečně přesunout na externí firmu.
Technologická architektura	Architektura aplikace je navržena tak, aby tutéž aplikaci ve stejné době mohly využívat tisíce uživatelů z mnoha různých organizací (víceuživatelská aplikace, Multi-Tenant Architecture). Technologicky je nutné zajistit segmentaci uživatelského adresáře (directory) a s tím související segmentaci databázového systému tak, aby bylo zajištěno, že	Architektura navržena pro situaci, kdy každou instanci aplikace využívají pouze uživatelé jedné organizace. Samostatná instance spuštěného kódu aplikace pro každou organizaci (tenanta) zajišťuje, že autentizovaní uživatelé dané organizace získají v rámci běhu dané	Architektura navržena pro situaci, kdy aplikaci využívají pouze uživatelé jedné organizace.

	autentizovaní uživatelé určitého tenantu (zákaznické organizace) získají přístup pouze do jim přiděleného segmentu databázového systému. Multi-tenantní aplikace: • Pro každého uživatele spouští samostatný proces operačního systému, který „běží samostatně“ po tomtéž kódu aplikace, • Pro každého uživatele si musí pamatovat, stav rozpracování transakce, tj. aby po návratu po přerušení procesu (způsobeném např. čekáním na odpověď uživatele), mohla pokračovat po ukončení přerušení v dalším kroku transakce přesně v místě posledního přerušení). Tzn. že multi-tenantní aplikace pracuje s mnoha stavy – musí si uchovávat vektor stavů samostatně pro každého uživatele. To single-tenantní aplikace nedělá. • Rozlišuje mezi daty, které sdílí všichni uživatelé a daty, které patří jen danému uživateli: ○ Data prvního typu jsou např. rezervace na sedadla v letadle v rezervačním systému aerolinek (když jeden uživatel sedadlo rezervuje, již se sedadlo nesmí nabízet jinému uživateli). Jinými slovy uživatelé se v tomto případě „perou“ o stejná data.	instance aplikace přístup pouze do vlastní instance databázového systému.	
--	--	--	--

	○ Data druhého typu jsou např. účetní záznamy v aplikaci účetnictví. Každý uživatel má jen svoje účetní data. • Zásadní rozdíl je také v řešení customizace. Jestliže určitá část kódu má u multi-tenantní aplikace pracovat jinak pro různé uživatele, musí být tyto odchylky uloženy všechny v kódu aplikace a stavový vektor určuje, se kterou odchylkou se má v dané chvíli pro daného uživatele pracovat. To je i důvod, proč customizace je v multi-tenantní aplikaci omezenější než u single-tenantní aplikace.		
Integrace aplikace s ostatními SW komponentami zákazníka	Integraci s komponentami provozní platformy zajišťuje poskytovatel v rámci ceny služby. Integraci s ostatními aplikacemi zákazníka, zejména těmi, které jsou provozovány v datovém centru zákazníka, resp. v datovém centru jiného poskytovatele je zodpovědností zákazníka.	Stejně jako v případě SaaS.	Integrace s komponentami provozní platformy i integrace s ostatními aplikacemi zákazníka jsou v zodpovědnosti zákazníka, který integraci většinou řeší přes externího dodavatele (systémového integrátora). Rozdíl oproti modelu SaaS je, že může docházet k porušení integrace při změně verzí jednotlivých komponent provozní platformy. Řešení těchto problémů má podstatně větší nároky na IT odborníky než v případě SaaS.
Uživatelské rozhraní aplikace	Forma řešení uživatelského rozhraní byla dříve (před deseti a více lety) významným rozlišovacím kritériem srovnávaných modelů, protože výběr formy má vliv dobu odezvy aplikace a na provozní náklady aplikace. Je-li použit pouze webový prohlížeč (lehký klient), pak to eliminuje potřebu vyvinout, na všech koncových zařízeních instalovat a podporovat složitější uživatelské rozhraní (těžký klient). V modelu SaaS bývá webový prohlížeč většinou jediným přístupovým mechanismem k aplikaci. V současné době je volba mezi lehkým a těžkým klientem většinou dostupná ve všech třech modelech. Je na zákazníkovi a jeho uživateli kterou z forem využijí.		

Funkcionalita pro řízení služeb	Aplikace má jako svoji součást „service management“, která umožňuje měřit, řídit a vyúčtovat dodávanou službu.	Může, ale nemusí být standardní součástí aplikace.	Nebývá standardní součástí aplikace. Administrace se někdy provádí nástroji uvnitř aplikace, případně nástroji třetích stran.
Údržba – nové verze SW (Upgrades)	Nové verze SW mohou být uváděny do provozu často a v menších inkrementech. Upgrade je realizován pouze provozovatelem. Všichni zákazníci obvykle využívají stejnou verzi a jsou převedeni na novou verzi ve stejný okamžik. To poskytovateli služby snižuje náklady provozu aplikace a současně urychluje vývojový cyklus aplikace. Někteří poskytovatelé umožňují po určitou dobu využívat i starší verzi aplikace.	Nové verze jsou obvykle nabízeny jednotlivým zákazníkům individuálně, a zákazníci jsou migrováni na novou verzi aplikace po dohodě a ve spolupráci s poskytovatelem. Zákazníci mohou využívat různé verze SW. Poskytovatel musí mít zaveden proces řízení verzí SW.	Nové verze jsou předávány uživatelům obvykle s frekvencí 12 – 24 měsíců. Zákazník je zodpovědný za instalaci nové verze. Zákazníci mohou využívat různé verze SW. Jak zákazník, tak i dodavatel (z důvodu vydávání hotfixů ke starším verzím) musí mít zaveden proces řízení verzí SW.
Rozhodnutí o nasazení nové verze SW	Defaultně o nasazení nové verze rozhoduje provozovatel; pouze v situaci, kdy je poskytovatel schopen poskytovat několik různých verzí multitenantní aplikace souběžně, je rozhodnutí o migraci na novější verzi rozhodnutím zákazníka (v rámci jistého časového okna). To, že zákazník za normálních okolností nemá nasazení nové verze pod svojí kontrolou, může mít negativní dopady na koncové uživatele a integritu IS.	Defaultně to je zákazník, avšak za jistých okolností to může být i provozovatel (pokud si společné upgrady pro všechny zákazníky vymíní ve smlouvě).	Zákazník.
Flexibilita a spolehlivost řešení			
Rychlost zavedení služby [8]	Služba je relativně rychle využitelná díky krátkému implementačnímu cyklu. Obvykle nevyžaduje u zákazníka instalovat žádný nový HW ani SW.	Stejně jak u SaaS.	Delší než u SaaS díky instalaci SW, HW a transferu znalostí potřebných pro provozování aplikace.

	Čas od rozhodnutí k zahájení užívání v řádech týdnů. U OVS, které musí postupovat dle ZZVZ, se musí počítat i s dobou nutnou na realizaci veřejné zakázky.		Čas od rozhodnutí k zahájení užívání v řádech měsíců. U OVS, které musí postupovat dle ZZVZ, se musí počítat i s dobou nutnou na realizaci veřejné zakázky.
Dostupnost služby	Služba je typicky dostupná 24 hodin 365 dní v roce z jakékoli lokality na světě, která je připojena k internetu.	Stejně jak u SaaS.	Přístup k aplikaci může být omezen pouze na koncová zařízení v lokalitách zákazníka nebo na zařízení, na kterých je instalován speciální klientský program. Provozní doba může být omezena (např. na pracovní dny).
„Škálovatelnost“ služby [8]	Objem odebírané služby (tj. počet podporovaných uživatelů, objem zpracovávaných dat apod.) může být měněn směrem nahoru i dolů (dynamická alokace zdrojů).	Stejně jak u SaaS.	V případě, kdy provozovatelem je zákazník, je technologická infrastruktura postavena na maximální zatížení a nemůže být pružně měněna. Zdroje nejsou optimálně vytěžovány.
Spolehlivost služby	Typicky velmi vysoká. Poskytovatel může efektivněji investovat do záložních technologií. Jestliže není schopen zajistit vysokou spolehlivost, ztrácí zákazníky.	Stejně jak u SaaS.	Zajistit vysokou spolehlivost dedikované infrastruktury je možné, ale nákladné. Proto se řada organizací spokojí s nižší spolehlivostí a akceptuje riziko výpadku, resp. prioritizuje pro běh na vysoce dostupné architektuře jen několik vybraných IS.
Funkcionalita aplikace	Rozsah funkcionality je obvykle dán většinovými potřebami určitého segmentu zákazníků. Multi-tenancy umožňuje přímý kontakt se všemi zákazníky (hodnocení vlastností služby, hlášení chyb, podávání požadavků na úpravy on-line) a urychluje cyklus aktualizací SW, zkvalitňuje transformaci znalostí, umožňuje poskytovateli analyzovat četnost využití	Různé varianty – může být stejné jako SaaS až po stejné jako on-premise. Rozlišujícím aspektem je zde zpravidla skutečnost, zda je aplikační SW skutečně poskytován formou služby (a zákaznický tenant je spravován poskytovatelem), nebo zda si zákazník správu tenantu zajišťuje sám; v takovém případě je přístup k provozním údajům pro poskytovatele a tím i znalost	Rozsáhlá funkcionalita, zákazník často využívá pouze její část (instalována je však obvykle celá funkcionalita). Statistiky frekvence využití funkcí a chybových stavů se obvykle neprovádějí. S dalším rozvojem technologií on-premise se však i v tomto modelu výrobci snaží do aplikací zabudovat statistické a

	nových funkcí a četnost chybových stavů. Pružnost poskytovatelů je lepší v případě, kdy existuje konkurenční nabídka služeb, kdy poskytovatelé jsou nuceni přizpůsobovat služby měnícím se požadavkům zákazníků.	způsobu využívání aplikačního SW pro poskytovatele omezený. Vyhodnocování frekvence využití jednotlivých funkcí aplikace i vyhodnocování chybových stavů aplikace je obtížnější při souběžném chodu většího množství instance dané aplikace.	reportovací nástroje, které výrobci hlásí zpět charakteristiky využívání a hlášení o chybových stavech.
Customizace (přizpůsobení specifickým požadavkům zákazníka) [8] [9]	Možná, ale značně limitovaná. Je to dáno podstatou modelu – všichni zákazníci využívají tutéž instanci SW. Objevují se nové nástroje (např.[12]), které umožňují upravit uživatelský interface, aplikační logiku i databázi.	Umožňuje rozsáhlou customizaci, neboť každý zákazník pracuje se svojí vlastní instancí aplikačního kódu. Pokud se zákazník a poskytovatel dohodnou na úpravách, je pro poskytovatele snadné novou verzi vyrolovat v cloudu k určitému termínu, případně ji dát k dispozici dříve pro účely testování.	Umožňuje rozsáhlou customizaci realizovanou úpravou kódu aplikace, avšak prakticky jen v případě aplikací vyvíjených na zakázku. Testování a nasazení upravených verzí je pak odpovědností zákazníka. V případě standardních („off-the-shelf“) aplikací není obvyklé získat individuálně upravenou verzi od výrobce, a zákazníci se musí spokojit s customizací změnou parametrů v rámci stávajícího kódu aplikace.
Změna poskytovatele služby	Zákazník si vytvoří Exit strategii, která popisuje kritéria hrubého porušení SLA, při jejichž naplnění zákazník spustí kroky k předčasnému ukončení smlouvy a k přechodu k jinému poskytovateli. Smlouva s poskytovatelem obecně popisuje i povinnosti poskytovatele po ukončení smlouvy (předání dat, spolupráce při migraci dat do nového prostředí apod.). Viz také nařízení EU Data Act 2023/2854, Smluvní ujednání týkající se změny poskytovatele, čl. 25 a 30.	Stejně jako u SaaS.	V případě, že zákazník má uzavřenou smlouvu o průběžných aktualizacích aplikace (typu „SW maintenance“), je předčasné ukončení smlouvy obvykle možné jen v případě, když SW má závažné chyby nebo jeho funkce jsou v rozporu s legislativou. Tento model je náchylnější na vendor lock-in, protože úroveň investice do technologické platformy i do znalostí IT personálu odpovědného za provoz aplikace je vysoká a může být bariérou přechodu k jinému poskytovateli.

Vyzkoušení aplikace uživateli	Aplikace může být (obvykle zdarma) vyzkoušena v testovacím provozu ve standardní verzi před zakoupením.	Uživatelé si aplikaci v podobě, kterou budou po nasazení využívat, mohou vyzkoušet až po zakoupení služby. Testovací provoz může být možný pouze s verzemi aplikačního SW, které jsou již k dispozici na základě customizací pro jiné zákazníky.	Uživatelé si aplikaci v podobě, kterou budou po nasazení využívat, mohou vyzkoušet až po zakoupení SW, jeho instalaci a customizaci. Testovací provoz je obvykle možný pouze s verzemi aplikačního SW, které jsou již k dispozici na základě customizací pro jiné zákazníky.
-------------------------------	---	--	--

Ekonomické aspekty			
Využití vnitřních zdrojů organizace (technologie, lidé)[8] [13][14]	Vyžaduje menší objem technologických a lidských zdrojů pro ICT podporu na straně zákazníka oproti on-premise modelu.	Stejně jako u SaaS.	ICT podpora provozu aplikace a její platformy z hlediska technologických a lidských zdrojů na straně zákazníka je výrazně náročnější než v případě SaaS.
Nákup SW licencí	Užití SW (tzn. i SW licence) je součástí ceny služby. Platí se pouze za služby, které byly v daném období objednány, resp. využity. Tj. snadná změna počtu SW licencí nahoru i dolů v měsíčním rytmu. Platí pro veřejný cloud, u privátního se licence kupuje.	V případě nákupu aplikace formou služby od poskytovatele je situace srovnatelná s SaaS. V případě, kdy zákazník kupuje licenci aplikace a sám ji provozuje (administruje) ve svém tenantu IaaS/PaaS, je situace srovnatelná s on-premise.	Nutno zakoupit samostatně SW licence. Stanovení správného počtu licencí může být obtížné. Změny nakoupeného počtu licencí jsou obvykle snadné pouze směrem nahoru; defaultně však nelze později vrátit nepotřebné licence a dostat zpět celou nebo i jen část nákupní ceny.
Náklady na pořízení a provoz aplikace – cena služby[8] [9][11] [17][13]	Nižší celkové náklady na aplikaci díky multiplicitnímu využívání těchto zdrojů a jednoduššímu modelu dodávky služby. Náklady jsou pro zákazníka jasně viditelné a predikovatelné, jsou rovnoměrně rozloženy v čase, nejsou investiční povahy. Díky škálovatelnosti služeb výše nákladů koreluje s objemem odebíraných služeb. Cena za službu je obvykle stanovena formou měsíčního poplatku (subscription fee), který zahrnuje cenu za provoz a údržbu aplikace v daném období. Cena se obvykle odvíjí od objemových charakteristik služby (počtu uživatelů, objemu zpracovávaných dat apod.). Při předplatném zákazník může platit i za	Obdobné jako v případě SaaS. S ohledem na to, že poskytovatel musí spravovat pro každého zákazníka samostatnou instanci SW, jsou náklady obvykle vyšší než v případě SaaS.	Řada nákladů je investiční povahy (nákup a instalace nového HW a SW). Existují náklady dané odepisováním investic. Celkové náklady nemusí korelovat s objemem využívaných ICT služeb (zdroje jsou dimenzovány na špičkové zatížení). Podstatná část výdajů je kumulována do období implementace SW, tj. před zahájením využití SW.

	nevyužité služby v situaci, kdy plný počet uživatelů služby nevyužívá.		
Finanční a pracovní náklady na klientské stanice	Na klientských stanicích plní funkci uživatelského rozhraní obvykle pouze webový prohlížeč (také nazývaný „tenký klient“). Změna aplikace nevyžaduje reinstalaci na všech klientských stanicích. Pokud existuje mobilní klientská aplikace (pro chytré telefony), instaluje a aktualizuje se zpravidla automaticky pomocí portálu organizace.	V případě, že pro přístup k aplikaci se používá webový prohlížeč, pak stejné jako v případě SaaS. V případě, že pro přístup k aplikaci se používá těžký klient, pak stejné jako v případě on-premise.	V případě „tlustého klienta“ změna aplikace obvykle vyžaduje reinstalaci na všech klientských stanicích. Trendem v oblasti on-premise je však využívání „tenkého klienta“ (webového prohlížeče) právě z důvodů snížení nákladů na údržbu klientské části. Mobilní aplikace pro on-premise SW obvykle nebývá k dispozici.
Druhy nákladů na službu	Pouze OPEX.	Primárně OPEX; pouze v případě nákupu licencí aplikačního SW (nikoli formou služby od poskytovatele) CAPEX.	Většinou CAPEX, pouze provozní náklady vnitřní infrastruktury a personální náklady jsou OPEX.
Aspekty řízení ICT organizace			
Motivace pro malé a střední organizace (OVS)	Malé a střední organizace mohou využívat stejně pokročilou technologii a funkcionalitu jako velké organizace.	Stejně jako u SaaS.	S ohledem na objem počátečních investic i nákladů na IT personál zvýhodňuje velké organizace před menšími.
Motivace dodavatele	Díky podstatě modelu je dodavatel motivován k udržování permanentní spokojenosti zákazníků (zákazník neudělal počáteční investici do licencí SW a ukončení subskripčního modelu je tak v zásadě jednodušší).	V případě, že zákazník platí za službu formou subskripce, je motivace dodavatele stejná jako v případě SaaS. V případě jednorázové platby za licenci aplikačního SW je situace srovnatelná s on-premise.	Zákazníkovi je věnována mimořádná péče zejména před nákupem licence. Po prodeji se zájem dodavatele přesouvá na další potenciální zákazníky.
Předmět kontraktu	Kontrakt má podobu smlouvy na dodávku informatické služby – SLA. Jeho hlavní části vymezují: <i>obsah služby</i> (funkcionalita, data, uživatelská podpora, ...), <i>objem služby</i> (počet uživatelů, objem transakcí, objem dat, ...), <i>kvalitu služby</i> (dostupnost, dobu	V případě, že zákazník platí za službu formou subskripce, je situace stejná jako u SaaS. V případě jednorázové platby za licenci aplikačního SW je situace srovnatelná	Kontrakt je obvykle rozdělen na několik dílčích smluv na HW, na SW licence a na doprovodné služby (implementace, integrace, školení, podpora, ...).

	odezvy, bezpečnost, ...) a <i>cenu služby</i> . Školení, integrace na ostatní aplikace zákazníka a podpora zákazníka při provozu jsou samostatné části smlouvy.	s on-premise (vyjma smlouvy na dodávku HW).	
SLA	Existence SLA je standardem. Uživatelé tak mají smluvně podloženo, jakou službu a za jakých podmínek dostávají.	Stejně jako u SaaS.	Je-li dodavatelem interní ICT útvar, není SLA ve většině případů používáno. Informatická služba tak obvykle není definována.
Zodpovědnost za ochranu dat před zneužitím a krádeží	Přenáší se na dodavatele v případě dat již uložených v systému. Za ochranu dat při přenosu může být smluvně zodpovědný poskytovatel připojení. Kvalita ochrany záleží na požadované bezpečnostní úrovni informačního systému a na konkrétních smlouvách o kvalitě služeb. V případě využití služeb zapsaných v katalogu cloud computingu (https://www.dia.gov.cz/oha/katalog-cloud-computingu/) má zákazník výhodu v tom, že poskytovatel i bezpečnost jeho služeb již byly prověřeny Digitální a informační agenturou, NÚKIB a dalšími státními orgány.	Stejně jako u SaaS.	Záleží na vnitřních pravidlech organizace, většinou je zodpovědný ICT útvar.
Využití ICT zdrojů [9] [11]	ICT zdroje poskytovatele (HW, SW, lidé) jsou využívány současně všemi zákazníky. Poskytovatel využívá úspory nákladů z rozsahu poskytovaných služeb. ICT zdroje na straně zákazníka jsou minimalizovány.	Stejně jako u SaaS.	ICT zdroje jsou využívány pouze jednou organizací, jejich využití v průběhu roku kolísá. Organizace musí disponovat i řadou IT specialistů (provozní pracovníci datového centra, správci HW a základního SW, pracovníci help desku, atd.), kteří nejsou u modelu SaaS zapotřebí.

ICT znalosti vyžadované na straně zákazníka	Potřebné znalosti: jak využít ICT služby na podporu činností organizace, ICT služby dostupné na trhu, obsah a struktura smluv na dodávku služeb, řízení dodávky služeb.	Stejně jako u SaaS.	Potřebné znalosti: totéž, jako v SaaS modelu a navíc široké spektrum technologických znalostí. Rozsah nutných znalostí je závislý na počtu používaných technologických platforem a počtu provozovaných aplikací. Pro získání a udržování těchto znalostí jsou třeba specializovaná školení.
Rozsah ICT personálu u zákazníka [14][15]	Velmi malý.	Stejně jako u SaaS.	Je-li provozovatelem zákazník, pak velký – jsou zapotřebí specialisté na různé technologické oblasti.
Procedury na řízení změn a zvládání incidentů	Smluvně daná doba reakce na incidenty, krátká doba realizace změny pro všechny zákazníky. Oprava chyby hlášené jedním zákazníkem znamená, že opravu v totéž okamžiku využívají všichni zákazníci. Jednoduchost této procedury snižuje náklady podpory uživatelů. Specifické změny požadované jedním nebo několika málo zákazníky jsou velmi problematické a někdy i neřešitelné.	Smluvně daná doba reakce na incidenty. Oprava chyby hlášené jedním zákazníkem může být poskytovatelem aplikována i do ostatních zákaznických tenantů, avšak s určitým zpožděním a s vyššími náklady na straně poskytovatele. Specifické změny požadované jedním nebo několika zákazníky jsou principiálně možné – záleží na konkrétních smluvních podmínkách.	Vzniklý problém je většinou řešen přes zprostředkovatele (implementátora, systémového integrátora). Opravy se řeší formou „SW záplat“, které si sám instaluje zákazník. Procedura je nákladná na straně zákazníka a bývá nespolehlivá.
Bezpečnost dat [8][9][11]	Relativně vysoká, definovaná smlouvou a bezpečnostními opatřeními na straně poskytovatele i na straně zákazníka. Vyšší plocha útoku (attack surface) mezi uživateli jednotlivých tenantů na úrovni aplikace, než v případě single-tenantního řešení nebo v případě on-premise.	Relativně vysoká, definovaná smlouvou a bezpečnostními opatřeními na straně poskytovatele i na straně zákazníka. Nižší plocha útoku mezi uživateli jednotlivých tenantů na úrovni aplikace než v případě SaaS.	Může být vysoká pouze za předpokladu, že jsou zavedena srovnatelná bezpečnostní opatření jako v případě cloudových služeb, zejména na úrovni IaaS/PaaS a v oblasti fyzické bezpečnosti datových center. V oblasti personální bezpečnosti získávají přístupová oprávnění obvykle pouze zaměstnanci

			organizace, avšak rozsah těchto oprávnění i délka jejich udělení bývá vyšší než v případě využívání cloudových služeb.
Rizika / kritické faktory úspěchu z pohledu zákazníka	Uživatel nemá přímou kontrolu nad svými daty – opírá se jen o smluvní zajištění a certifikace poskytovatele od třetích stran. Ztráta znalosti, která může být potřebná v budoucnu při rozhodnutí k návratu k on-premise modelu. Výběr vhodného poskytovatele služby. Stabilita poskytovatele. Optimální řešení Exit strategie. Úpravy aplikace jsou mimo kontrolu. Nasazování nových verzí je mimo kontrolu. Nedostatečná customizace aplikace. Nezajištěná integrace s ostatními aplikacemi. (Kdo má být zodpovědný za integraci?). Spolehlivost připojení k aplikaci.	Uživatel nemá přímou kontrolu nad svými daty – opírá se jen o smluvní zajištění a certifikace poskytovatele od třetích stran. Ztráta znalosti, která může být potřebná v budoucnu při rozhodnutí k návratu k on-premise modelu. Výběr vhodného poskytovatele služby. Stabilita poskytovatele. Optimální řešení Exit strategie. Nezajištěná integrace s ostatními aplikacemi. (Kdo má být zodpovědný za integraci?). Spolehlivost připojení k aplikaci.	Technologická zaostalost. Vysoké celkové náklady na ICT. Nízká flexibilita ICT služeb. Závislost na znalostech interního IT; případný insourcing specifických služeb je nákladný. Obtížná škálovatelnost. Úpravy aplikace pomocí smluvních dodatků. Otázka záruk rozvoje aplikačního SW, zakoupeného prostřednictvím jednorázových licencí.

5. Význam použitých zkratk

CC	Cloud computing
eGC	eGovernment Cloud
IaaS	Infrastructure as a Service, IT infrastruktura jako služba
ISVS	Informační systém veřejné správy
Materiální poskytovatel	Poskytovatel, který službu cloud computing produkuje. Materiální poskytovatel může mít řady přeprodejců, kteří služby materiálního poskytovatele přeprodávají koncovým zákazníkům.
Multi-tenantní software	Software, jehož jedna instance uložená v operační paměti poskytuje funkcionalitu uživatelům více zákazníků současně.
OVS	Orgán veřejné správy
PaaS	Platform as a Service, IT platforma jako služba
SaaS	Software as a Service, aplikační software jako služba
Single-tenantní software	Software, jehož jedna instance uložená v operační paměti poskytuje funkcionalitu uživatelům jednoho zákazníka.
SLA	Service Level Agreement – smlouva o úrovni poskytovaných služeb
ZKB	Zákon č. 181/2014 Sb., o kybernetické bezpečnosti
ZoISVS	Zákon č. 360/2000 Sb., o informačních systémech veřejné správy
ZZVZ	Zákon č. 134/2016 Sb., o zadávání veřejných zakázek

6. Použité zdroje

Definice a výklad SaaS:

- [1] Gartner Group: Definition of Software as a Service (SaaS), Gartner Information Technology Glossary: <https://www.gartner.com/en/information-technology/glossary/software-as-a-service-saas>. Gartner předpokládá „1 to many model“ (all contracted customers), pro common code and data.
- [2] Wikipedia: https://en.wikipedia.org/wiki/Software_as_a_service. SaaS je typicky multitenant, ale uvádí se také „but sometimes they offer a siloed environment for an additional fee.“
- [3] IBM: <https://www.ibm.com/topics/saas>. Uvádí se v detailu, že „SaaS applications use multi-tenant architecture, in which a single instance of the application serves every customer.“
- [4] Investopedia: <https://www.investopedia.com/terms/s/software-as-a-service-saas.asp>. V celkovém popisu uvádí: The software is provided on a subscription basis; The software is located on external servers rather in-house; Software as a service is commonly accessed through a web browser. Následně uvádí v detailu: ... network-based access to a single copy of an application that the provider created specifically for SaaS distribution.

Srovnání modelů:

- [5] IBM: <https://www.ibm.com/topics/multi-tenant>. Uvádí se, že „single instance of a software application (and its underlying database and hardware) serves multiple tenants“
- [6] Geekflare: Understanding Single Tenant vs. Multi-Tenant in Cloud Computing, <https://geekflare.com/single-tenant-vs-multi-tenant-cloud-computing/>. Podrobné srovnání vč. architektury; uvádí se: „SaaS providers use multi-tenant architecture to provide services to multiple customers“.
- [7] Workos: <https://workos.com/blog/singletenant-vs-multitenant>. V detailech se zmiňuje „Mixed-tenancy architecture“ a připouští se, že SaaS může mít různé modely, viz část „For SaaS apps, the multi-tenant architecture is almost always the better choice — you can deploy a single instance of your app and serve a huge number of customers, from 1 to 10,000. Adding more features to your app is also easier since you only need to push changes to a single instance. There are however some cases you'll want to go with a single-tenant architecture — when security regulations and data protection laws say so and when you want to support self-hosting.“

- [8]Itsvit: <https://itsvit.com/blog/single-tenant-vs-multi-tenant-clouds-which-is-right-for-you/>. Poskytuje podrobné srovnání výhody / nevýhody single- vs. multi- tenant; také uvádí „mixed tenancy models“.
- [9]Cloudzero: <https://www.cloudzero.com/blog/single-tenant-vs-multi-tenant/>. Zmiňuje se rovněž „Mixed tenancy model“.
- [10]Wikipedia: <https://en.wikipedia.org/wiki/Multitenancy>. Zde jde spíše o klasické pojetí multitenancy, a zmiňuje i „multi-instance architecture“
- [11]Stallions Solutions: <https://stallions.solutions/azure-single-tenant-vs-multi-tenant/>. Nabízí srovnání vlastností; nezmiňuje smíšené modely.
- [12]<https://www.salesforce.com/eu/?ir=1>
- [13]Voříšek, J. a kol.: Principy a modely řízení podnikové informatiky, Praha, Oeconomica, 2008, 2011, 2015, 446 s. ISBN 978-80-245-1440-6
- [14]Voříšek, J., Pour, J. a kol.: Management podnikové informatiky, Professional Publishing, Praha 2012, ISBN 978-80-7431-102-4
- [15]Voříšek, J. Enterprise information system development and operation principles. 1st ed. Prague: Wolters Kluwer, 2014. ISBN 978-80-7478-569-6
- [16]Voříšek, J., Pour, J. Buchalcegová, A.: Model for management of business informatics – principles and practices, E&M Economics and Management, 2015, Vol 18, No. 3, pp. 160-173, Wolters Kluwer, ISSN (Print) 1212-3609, ISSN (Online) 2336-5604
- [17]COUNCIL, S.-A.-A.-S. E.: Software & Information Industry Association, 2006. "Software-as-a-Service; A Comprehensive Look at the Total Cost of Ownership of Software Applications". http://www.webex.com/pdf/Total_Cost_of_Ownership.pdf
- [18]Gartner Report on Cloud Costs: Gartner pravidelně vydává rozsáhlé zprávy o nákladech na cloudové služby, které mohou sloužit jako výchozí bod pro srovnání viz např.: <https://www.gartner.com/en/infrastructure-and-it-operations-leaders/topics/cloud-computing>, <https://services.global.ntt/en-us/campaigns/gartner-predicts-2023>
- [19]AWS Whitepaper on Scalability: Amazon Web Services, https://docs.aws.amazon.com/whitepapers/latest/building-scalable-secure-multi-vpc-network-infrastructure/welcome.html?did=wp_card&trk=wp_card , nabízí mnoho whitepaperů, které se detailně zabývají různými aspekty cloudových služeb, včetně škálovatelnosti.
- [20] Azure SLA: Microsoft Azure, <https://www.microsoft.com/licensing/docs/view/Service-Level-Agreements-SLA-for-Online-Services?lang=1> poskytuje záruky dostupnosti (SLA) pro své služby, což je důležitý ukazatel spolehlivosti.
- [21]NIST Cybersecurity Framework: <https://www.nist.gov/cyberframework>, Tento rámec poskytuje standardy pro řízení kybernetické bezpečnosti, které lze aplikovat na různé typy systémů.

- [22] Forrester Wave: Cloud Platforms: <https://www.forrester.com/press-newsroom/the-forrester-wave-eight-global-public-cloud-platforms-and-how-they-stack-up/>, Forrester pravidelně vydává zprávy, které hodnotí různé cloudové platformy a jejich schopnosti.
- [23] Google Cloud Documentation: <https://cloud.google.com/support/docs>, Dokumentace Google Cloud poskytuje podrobný popis služeb a možností správy.
- [24] DevOps Handbook: <https://www.amazon.com/DevOps-Handbook-World-Class-Reliability-Organizations/dp/1950508404>. Tato kniha popisuje principy a postupy DevOps, které mohou urychlit vývoj a nasazování aplikací.
- [25] AWS Well-Architected Framework: https://aws.amazon.com/premiumsupport/business-support-well-architected/?trk=28d90059-d618-4962-a206-4b98af48e131&sc_channel=ps&s_kwid=AL!4422!10!72086999239241!72087548760101&ef_id=c2307a0412de185f43d61d3de70ccc26:G:s&msclkid=c2307a0412de185f43d61d3de70ccc26. Tento rámec poskytuje soubor best practices pro návrh a provoz robustních a bezpečných aplikací v cloudu.