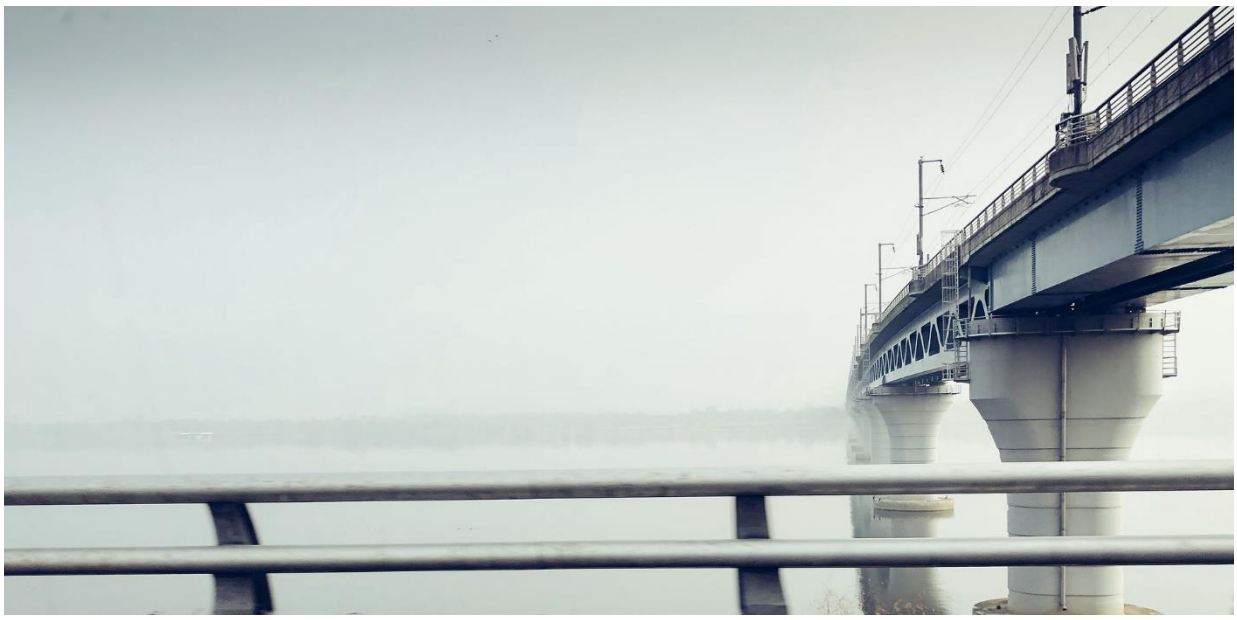
OBJECT GEARS



Průvodce návrhem CMDB řešení



V tomto dokumentu popisujeme, jak začít s procesem **Configuration management** a **Konfigurační databází** (CMDB – Configuration Management Database), s důrazem na jasnost a praktické příklady. Jde o obecný návod s ukázkou některých konkrétních aspektů, tak jak je řeší Konfigurační databáze platformy ObjectGears.

Business kontext **3**

Co je to Configuration management **3**

Rozsah CMDB **3**

Přínosy a návaznost na další procesy **4**

Change management 4

Release management 4

Incident a Problem management 4

Asset management 5

Project management 5

Informační bezpečnost 5

Jak začínáme **5**

Kulturní aspekt **6**

Tvorba modelu **6**

Položky, s nimiž začneme **7**

Rozdělení konfiguračních položek do vrstev **8**

Identifikátor a název položky **9**

Konkrétní aspekty hlavních typů konfiguračních položek **13**

Služby 13

Aplikace 13

Rozhraní (Interfacy) 14

Datové zdroje 14

Servery 14

Zařízení (Hardware) 15

Automatická vs. manuální aktualizace data **15**

Baseline **15**

Propagace manuálně nastavených dat do systémů **15**

Vizualizace dat **16**

Chybějící kritická data **16**

Shrnutí **17**

Business kontext

V dnešní době, kdy je moderní business tolik poháněn IT a IT se ve firmách stává vlastně businessem, je **transparentnost IT infrastruktury** kritickou záležitostí. Jak IT infrastruktura roste a stává se komplexnější, čelíme víc a víc výzvám konsolidace, údržby a porozumění komplexním informacím o konfiguraci.

Řešením je **konfigurační databáze**, která ukládá informace o IT infrastruktuře, ale nejen o ní – také o business prostředí / kontextu, a umožňuje klíčovým funkcím IT Service Management (ITSM) a dalším procesům používat konkrétní objekty těchto procesů – konfigurační položky.

Prakticky každá organizace v každém odvětví musí vyhovět regulatorním požadavkům jako je Sarbanes–Oxley (SOX), Health Insurance Portability and Accountability (HIPAA) nebo Payment Card Industry Data Security Standard (PCI DSS). Další regulace vytvářejí jednotlivé vlády nebo nadnárodní organizace. Jedním z cílů těchto regulací shodně bývá zajistit bezpečnost systémů a citlivých dat, jejich odpovídající řízení a odpovědnost za tato data a systémy. Konfigurační databáze je základním nástrojem pro audit a zajištění compliance.

Co je to Configuration management

ITIL V3 popisuje Configuration management jako:

„Proces zodpovědný za zajištění toho, že aktiva potřebná k dodávání služeb jsou správně řízena a že o těchto aktivech máme přesné a spolehlivé informace, kdykoli je to potřeba. Tyto informace obsahují detaily nastavení a vazeb mezi těmito aktivy.“

Konfigurační databáze je předpokladem správného fungování tohoto procesu, základnou, nad níž tento proces může běžet. Configuration managementem se přitom zabývají také další mnohé standardy. V níže uváděných příkladech budeme pracovat s typickými konfiguračními položkami IT. Stejně tak bychom ale mohli řešit například evidenci výrobních zařízení nebo jakýchkoli používaných technologií s cílem zajistit jasnost v otázce produktů, služeb a toho, že budou mít (pro svého příjemce/zákazníka) žádoucí a očekávané vlastnosti a kvalitu.

CMDB modeluje vaše služby, aplikace a infrastrukturu. Jde o systém podporující správné rozhodování v IT, ale má přesahy do dalších oblastí včetně **auditu, compliance či zákaznického servisu**.

Rozsah CMDB

Každá organizace řeší adekvátní rozsah CMDB. Snaha obsáhnout všechny možné položky a aspekty bude nákladná, náročná na údržbu a čas, těžkopádná, a proto pravděpodobně odsouzená k neúspěchu. Naopak model nesoucí příliš málo informací nedosáhne potřebných zlepšení. Níže si proto ukážeme i jak stanovit optimální úroveň evidence. Pokud na začátku identifikujeme základní entity, u kterých pociťujeme nejvyšší potřebu evidence, a ty implementujeme, jsme na dobré cestě. Dále je vhodné si vytvořit plán, podle kterého budeme postupovat.

Přínosy a návaznost na další procesy

Institute of Configuration Management (icmhq.com) ve své studii Monetary Value of Traditional Configuration Management versus CMII (publikováno 14. ledna 2014) odhaduje, že náklady na prodej lze snížit až o 40% právě implementací Configuration managementu v důsledku minimalizace nejrůznějších nápravných opatření, která nyní firmy běžně musí činit. Pokud bychom hledali příklady těchto potenciálních úspor v IT, představme si tyto situace.

1. Pokaždé, když zvažujeme určitou změnu, scházíme se se správci daných systémů a probíráme souvislosti. Vše vysvětlit opakovaně zabírá čas. (V případě implementace Konfigurační databáze bychom zobrazili položky, které chceme vidět, a hned pochopili dané souvislosti.)
2. Pokud implementujeme změnu do určité aplikace, zjistíme až v průběhu prací, že původně předané informace nebyly úplné (existují např. další systémy, s nimiž je aplikace propojena a které jsou změnou dotčené) nebo přesné (používá se např. jiná verze aplikace). Dodatečné změny jsou nákladné a opravujeme něco, co jsme již jednou, nesprávně, udělali. Stojí nás to zbytečné peníze a ztrácíme čas.
3. Chceme obnovit smlouvu o údržbě a servisu hardware s dodavatelem. Musíme se vydat s kolegy do serverovny, abychom si byli jistí, jaký hardware přesně používáme, protože víme, že na naši stávající evidenci se nemůžeme spolehnout. Následně budeme řešit, u jakého hardware potřebujeme údržbu na celý rok a kde jen na část, protože je plánovaná migrace na novější hardware. Chceme si být jistí, že na každý hardware platíme adekvátní typ podpory (kratší lhůty na odezvu dodavatele v případě problému znamenají vyšší poplatky za servis). Nevíme však, co vše na jednotlivých serverech běží, jaké aplikace a jak jsou kritické. Nakonec zaplatíme podporu na všechny servery za celý rok a raději vyšší úroveň podpory, protože nejsme schopni vše dohledat. (V případě Configuration management procesu si vyfiltrujeme servery, u nichž nám bude končit podpora. Potřebnou úroveň servisu si zjistíme z kritičnosti aplikací, které na nich běží.)

Configuration management a konfigurační databáze úzce souvisí s dalšími procesy. Níže je několik příkladů.

Change management

Ve chvíli, kdy plánujeme určitou změnu, je vhodné odkazovat na konkrétní položku z konfigurační databáze a být schopni zobrazit další změny, které s ní počítají a zobrazit závislosti. Vyhneme se tak protichůdným aktivitám a z vazeb dané položky odhadneme dobře všechny dopady.

Release management

Ve chvíli, kdy plánujeme release nové aplikace, je dobré jasně označit položky, kterých se release bude týkat, jak pro schválení releasu, tak pro zajištění, že se nasadí vše, co je potřeba.

Incident a Problem management

Pokud dojde k výpadku nebo omezení určité služby, můžeme snadno identifikovat položky, u nichž došlo v poslední době k nějaké změně a které pravděpodobně budou příčinou výpadku. To nám pomůže incident rychleji vyřešit.

Pokud zjistíme, že na dané položce jsme incident řešili již opakovaně (konfigurační položka byla opakovaně napojena na určitý incident), bude nám jasné, že bychom se příčinou měli zabývat dále, a to v procesu Problem management.

Asset management

Řada konfiguračních položek je aktivy, které řídíme z pohledu správy majetku. IT poskytuje informace o stavu, lokaci a změnách týmům Financí. Jedná se o zařazení do majetku, pravidelné inventury, otázky technického zhodnocení a vyřazení z evidence.

Project management

Projekty a projektoví manažeři mohou získat z CMDB velmi cenné informace. Už ve fázi studie proveditelnosti a schvalování projektu je vhodné stanovit, které konfigurační položky budou dotčeny a jak. Pokud to učiníme, výrazně zvýšíme kvalitu těchto studií a přesnost odhadu nákladů. Budeme také vědět, které technické a business týmy budeme v projektu potřebovat. Bez těchto kalkulací riskujeme zpoždění projektu a neočekávané náklady v průběhu implementace projektu. Výstupy projektu by také měli být reflektovány v konfigurační databázi. Proto by měly vzniknout závazné postupy stanovující interakci projektů a Configuration managementu, se kterými budou všichni dobře seznámeni a budou se jimi řídit.

Informační bezpečnost

Konfigurační databáze poskytuje snadný přístup k zásadním informacím pro posouzení bezpečnosti informačních systémů, rizik a zranitelností. Tým bezpečnosti informací by měl také posoudit rizika vyplývající z přístupu jednotlivých rolí pracující v konfigurační databázi s jednotlivými konfiguračními položkami. Citlivá data nacházející se v konfigurační databázi mohou být např. IP adresy, úrovně patchů, komunikační protokoly, jména administrátorů a jejich telefonní čísla.

Jak začínáme

Pokud uvažujeme o implementaci konfigurační databáze, měli bychom si zmapovat naši výchozí situaci. Možná nyní evidujete data o aplikacích, serverech nebo hardware v tabulkách Excelu. Pokud tomu tak skutečně je, jste na dobré cestě. I když si myslíte, že evidovat takováto data v Excelu je projevem nedostatečné profesionality, jde o to, že si již uvědomujete důležitost této evidence, máte data, byť třeba ještě neúplná a v nedostatečné kvalitě, ale to je něco, na čem se dá zapracovat. Již se snažíte tento proces zlepšit. Data je možné z Excelu snadno uploadovat do Konfigurační databáze a pak využít její reporting pro identifikaci chybějících klíčových dat.

Pokud data zatím nemáte a dané informace se nacházejí pouze v hlavách jednotlivých pracovníků, je Excel dobrým startovním bodem. Pomůže nám ujasnit si, co u kterých položek chceme evidovat a jaké vztahy mezi sebou budou mít. Zároveň se budeme muset vypořádat s kulturním aspektem zavádění Configuration management.

Kulturní aspekt

Při zavádění konfigurační databáze budeme řešit situaci, kdy budeme chtít evidovat určité informace. Ty ale budeme muset nejprve získat od kolegů, kteří jsou za dané konfigurační položky zodpovědní. Informace znamenají moc, a proto nemusí být vždy lehké je získat. Jak uspět?

Podpora od managementu a formální schválení projektu

Pojměte **zavádění Konfigurační databáze jako projekt**: jeho přínosy vysvětlíte managementu, jehož podporu získáte a jemuž budete pravidelně reportovat postup na projektu. Vytvořte zakládací listinu projektu, kde vysvětlíte, jaké stávající problémy chcete řešit, jakých cílů chcete dosáhnout a v jakém horizontu, jaké zdroje budete potřebovat (lidé, finance) a proč se to vyplatí (business case popisující finanční i nefinanční přínosy – např. minimalizaci rizik daných špatným stávajícím stavem informací o konfigurační základně, řešení auditního nálezu atd.). Pokud se setkáte s problémy při dosažení cílů, které management schválením projektu stanovil, např. v oblasti aktualizace chybějících dat, danou věc přednesete ve zprávě o postupu projektu jako *issue* včetně návrhu na jeho řešení: např. manažer týmu aplikačních administrátorů dostane za úkol doplnit chybějící data o komponentách aplikací a výměně dat (rozhraní). Podpora od managementu včetně formálního schválení vašeho záměru, na které se budete moci vždy odvolat, je velmi důležitá. Jako u každého projektu byste měli mít jednoho (max. dva) sponzory – manažery z vedení, kteří za vámi budou stát a kteří projekt opravdu chtějí prosadit.

Nakoupení uživatelů

Každá změna vyvolá obavy či odpor. Jde o přirozenou lidskou reakci. Věnujte čas vysvětlení, proč se do této aktivity pouštíte. Připravte si příklady, na kterých vysvětlíte určité scénáře – stávající problémy a jak je chcete řešit v budoucnu. Ty se vám již jistě hodili při přesvědčování vedení. Teď vás čeká přesvědčování pracovníků středního a nižšího managementu i koncových uživatelů. Při schůzkách se setkáte s námitkami, na které můžete, ale nemusíte mít hned jasnou odpověď. Je naprosto v pořádku si námitky poznačit a vrátit se s odpovědí někdy později. Možná vás nebude bavit poslouchat, že takových projektů už byla spousta nebo že to nemá smysl, a věřte, že se s takovými připomínkami setkáte. I takové podněty si však zaslouží odpověď. Připomenutí, že o realizaci projektu rozhodlo vedení organizace je tou správnou reakcí na tento typ rezistence některých spolupracovníků.

Úspěch projektu zavedení CMDB

Úspěch zavedení CMDB a procesu Configuration management závisí na přidané hodnotě, kterou budou uživatelé vnímat. Pokud lidé nevidí přínosy, s výstupy projektů nepracují a celá iniciativa je odsouzena k neúspěchu. Toto je třeba nepodcenit. Zároveň je pravda, že implementace konfigurační databáze představuje ohromný potenciál ve většině středních a velkých organizací. Je na nás tento potenciál využít a dobrat se úspěchu.

Tvorba modelu

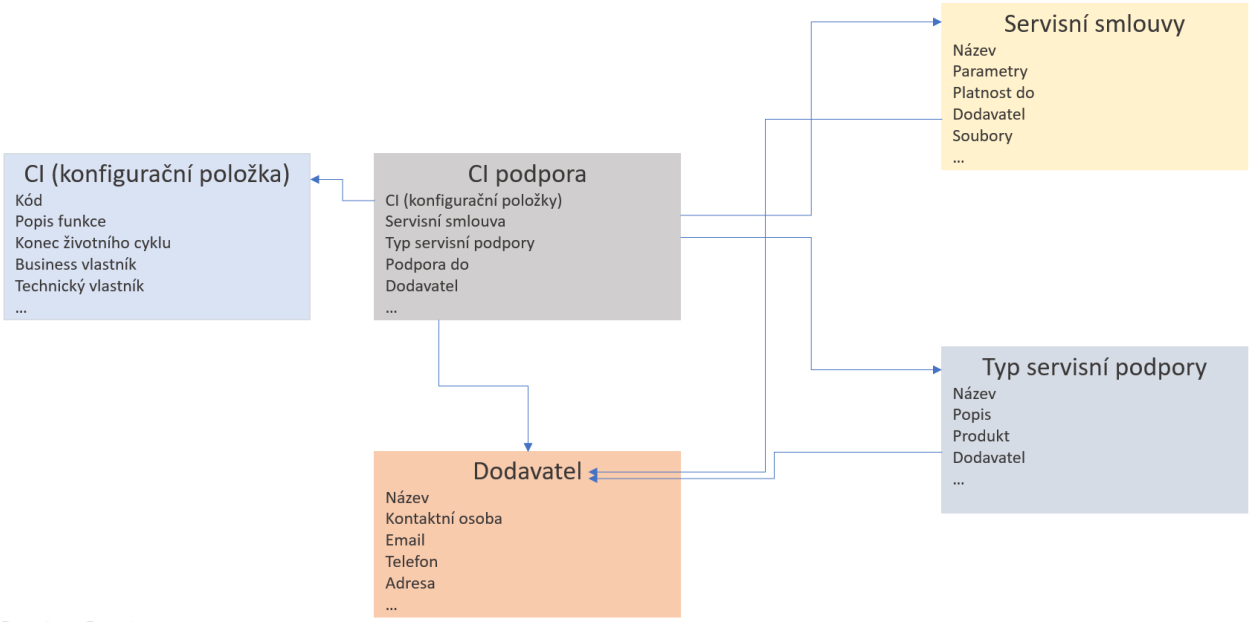
Pokud se rozhodnete pro implementaci konfigurační databáze, pořídíte si ji pravděpodobně již s nějakým modelem obsahujícím různé typy konfiguračních položek, s různými vlastnostmi a různými vzájemnými

vztahy, něčím, co výrobce dané konfigurační databáze označuje jako *best practise*, alespoň v rámci jeho zkušeností.

Důležité je si uvědomit, že **neexistuje jeden optimální model uspořádání** těchto **konfiguračních položek**. Existují různé organizace různých odvětví a různých velikostí. Ani v rámci jedné konkrétní firmy nelze říci, že existuje jeden optimální model. Ve skutečnosti záleží i na tom, kdo adekvátnost modelu bude posuzovat, jaké má zkušenosti s danou problematikou, s čím se setkal a co již někdy řešil. Tyto osobní pohledy mohou vést k odchylkám a úpravám modelu. Není to ale většinou nic zásadního. Důležitější pro úspěch celého projektu je celková identifikace uživatelů s daným modelem, jejich buy-in, než víceméně akademické spory, zda se daná informace má evidovat jako vlastnost určité položky či jako záznam ve zvláštní třídě. Model nemusí být statický a v budoucnu ho můžeme přehodnotit a změnit. Samozřejmě, že si takovéto změny vyžádají určitou pracnost, ale je lepší projekt úspěšně rozjet než se na samém začátku zastavit, protože lidé nepřijmou něco, co jim přijde moc složité.

Model si tedy projděte s klíčovými pracovníky, potvrďte si, že jim vše dává smysl, a udělejte úpravy, které budou třeba.

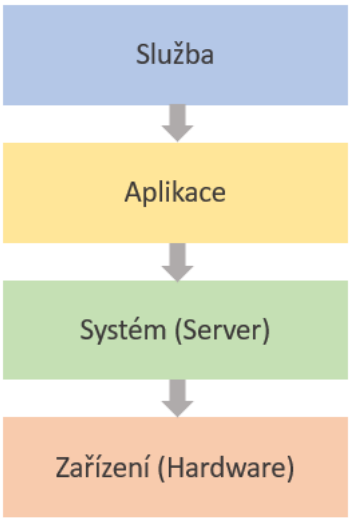
Příklad: U serverů je vhodné sledovat pokrytí servisní smlouvou. Konfigurační položku Server můžeme doplnit o vlastnost Servisní smlouva (prostý text), kterou budeme schopni rychle vyplnit. Pokud budeme chtít data evidovat lépe, můžeme použít řešení s několika třídami, které bude obecnější a obsáhne mnohem více strukturovaných informací. Viz následující obrázek.



Obrázek 1: Servisní smlouvy nám umožní evidovat potřebné informace o podpoře konfiguračních položek.

Položky, s nimiž začneme

Konfigurační databáze může běžně mít desítky či stovky typů položek. Následující schéma představuje základní rozdělení hlavních konfiguračních položek.



Obrázek 2: Základní rozvrstvení konfiguračních položek.

Služby představují zajištění konkrétní potřeby businessu, např.: *Zavést nového business partnera/zákazníka/dodavatele* nebo *Vystavit objednávku*.

Služby jsou často realizovány aplikacemi, v nichž uživatelé provádějí úkony k realizaci dané služby.

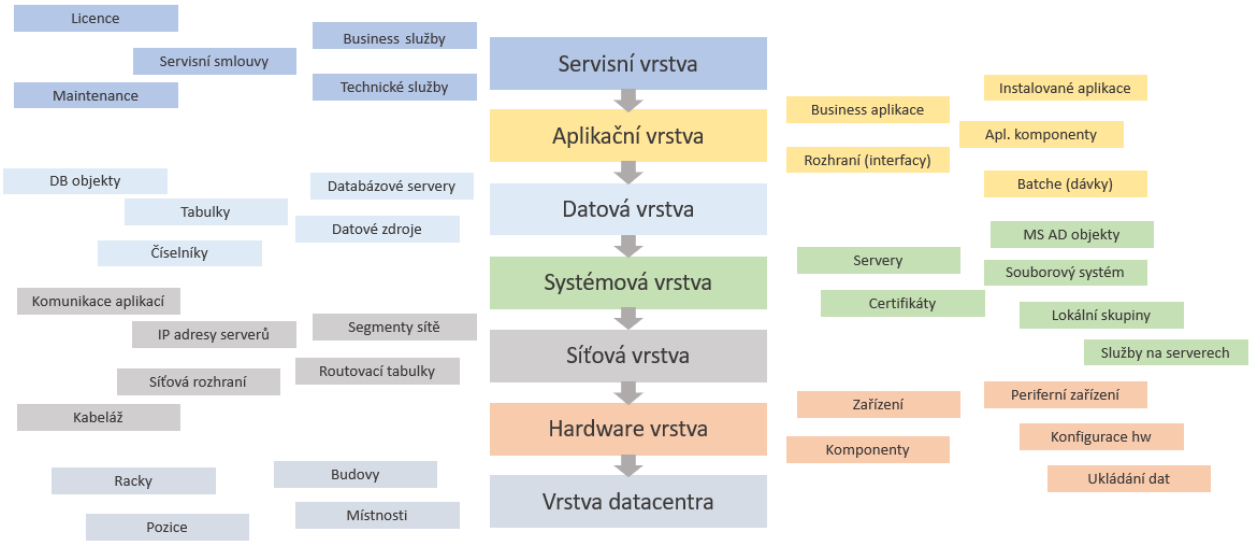
Aplikace běží na systémech, serverech s operačním systémem.

Tyto systémy, servery, potřebují zařízení (hardware), na němž běží. Model by měl být připraven zachytit i odlišné situace dané virtualizací a používáním cloudových služeb.

Dobré je identifikovat ty položky, s kterými chceme začít. Pomůže nám to dosáhnout výsledků a vyzkoušet si postup, pomocí nějž model dále rozvineme. Je vhodné začít např. s následujícími entitami: Zařízení (hardware), Machine (OS server), Databáze, Aplikace, Rozhraní (Interfacy), Služby.

Rozdělení konfiguračních položek do vrstev

V ObjectGears máte možnost pracovat s modelem dle vaší volby. Výchozí model člení položky do následujících hlavních vrstev, které mohou obsahovat další podvrstvy.



Obrázek 3: Detailnější rozvrstvení konfigurační položek.

Při prvotní implementaci nebudete asi řešit většinu položek. Můžete je skrýt a aktivovat ve chvíli, kdy je začnete používat.

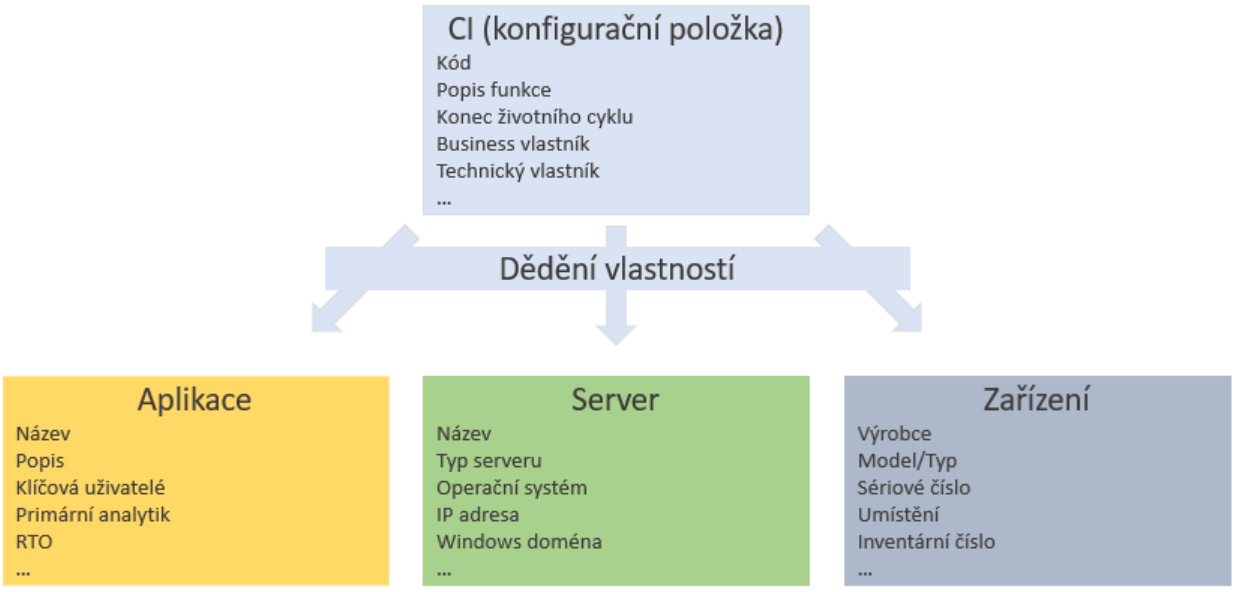
Identifikátor a název položky

Unikátní a neměnný název konfigurační položky?

Někdy se můžete setkat s názorem, že každá konfigurační položka by měla mít jedinečný název, který by se neměl měnit v průběhu jejího životního cyklu. Ve skutečnosti se zde míchá několik věcí dohromady. Níže se pokusíme vysvětlit, proč je tento koncept chybný a jak se s tímto tématem vyrovnat lépe a praktičtěji.

Vlastnosti položky

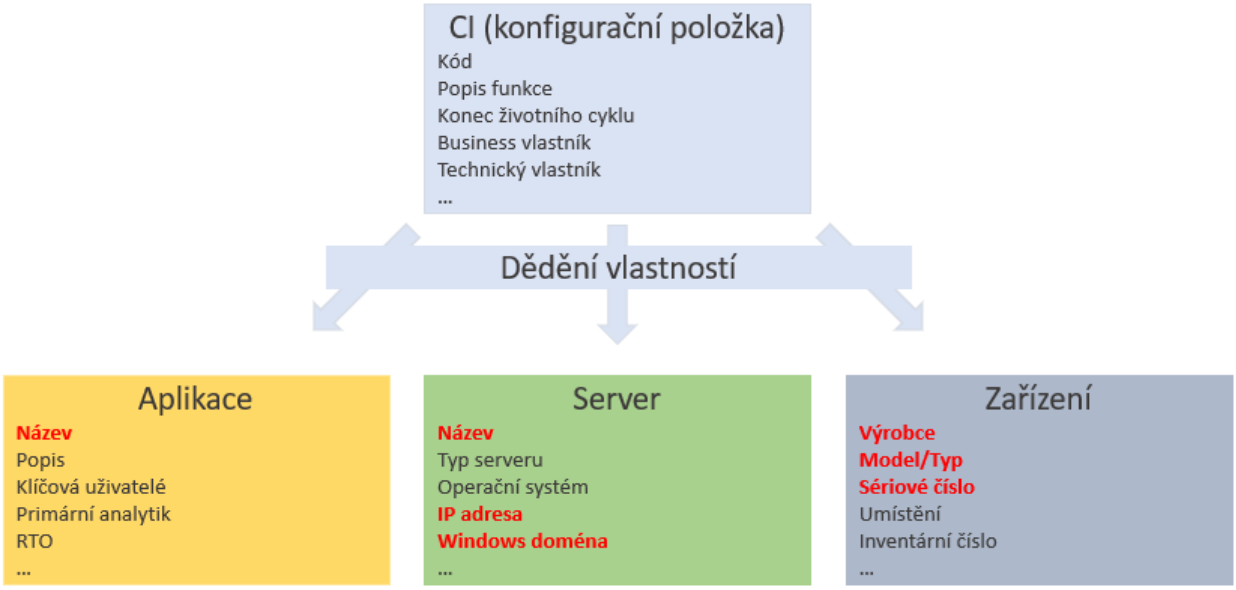
Konfigurační položku reprezentují její vlastnosti, které se rozhodneme sledovat v konfigurační databázi. Tyto vlastnosti mohou být společné všem konfiguračním položkám (např. unikátní kód položky, který budeme generovat, popis funkce, konec životního cyklu nebo business vlastník položky, který schvaluje změny a technický vlastník, tedy administrátor, který ji podporuje a implementuje změny). Další vlastnosti mohou být specifické pro daný typ položky. Jiné vlastnosti budou mít hardware zařízení, jiné databázové servery, jiné aplikace a jiné třeba certifikáty.



Obrázek 4: Jednotlivé typy konfiguračních položek dědí společné vlastnosti ze společného předka.

Identifikující vlastnosti

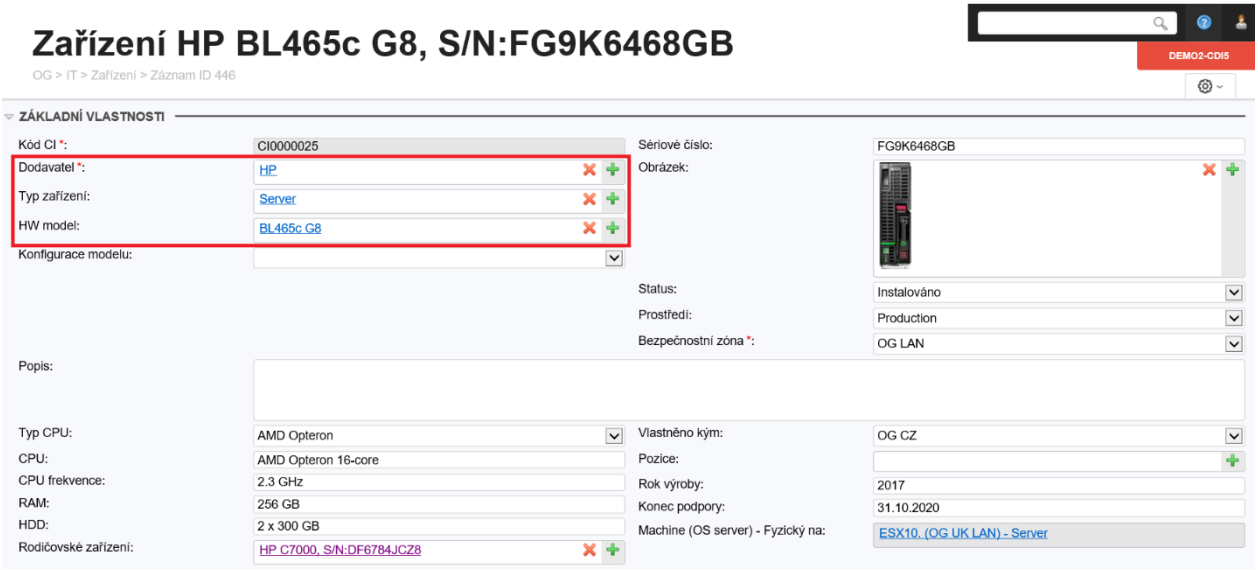
Položky musíme být schopni rozlišit. Každý typ položky by měl mít určité vlastnosti, jejichž kombinace bude unikátní, tzn., že podle nich okamžitě poznáme, o jakou položku jde. Pokud takovéto vlastnosti v modelu zatím nemáme, měli bychom se nad naším modelem zamyslet a vylepšit ho. V případě Zařízení takovouto unikátní kombinací budou hodnoty vlastností Výrobce zařízení, Typ/model a Sériové číslo. V případě serverů nás možná napadne jediná vlastnost: plné doménové jméno. Když se ale zamyslíme a vezmeme v úvahu, že některé servery mohou být mimo doménu a mohou existovat ve dvou oddělených sítích klidně i se stejným jménem, je nám jasné, že budeme potřebovat další identifikaci. Tou může být IP adresa nebo segment sítě. Ty se však mohou měnit. Podobně u aplikací za identifikátor budeme považovat název aplikace. Někdy ale může být dobrý důvod název aplikace změnit. Přesto se bude nadále jednat o tu samou konfigurační položku.



Obrázek 5: Identifikující vlastnosti konfiguračních položek jsou vyznačeny červeně.

Neměnný a jedinečný identifikátor

Pokud někde zmiňujeme nějakou konfigurační položku, měli bychom zajistit, že i po změně určitých vlastností, ke kterým přirozeně dochází, stále bude jasné, o jakou položku jde. Pokud na položku odkazujeme v rámci konfigurační databáze, tato vazba je zajištěna odkazem (cizím klíčem). Pokud položku zmiňujeme někde jinde, měli bychom použít její kód (např. CI00040970). Všimněte si, že kód nenese žádnou informaci o názvu, výrobci a podobně. Z kódu pouze poznáme, že jde o konfigurační položku (CI – Configuration Item). Jde o unikátní kód, který automaticky přidělí konfigurační databáze všem položkám, a je to jedna z vlastností společná všem typům konfiguračních položek.



Obrázek 6: Příklad konfigurační položky (zařízení) – identifikující sloupce jsou zvýrazněny.

Popis položky

Dále budeme potřebovat popis položky. Jde o obdobu názvu zmiňovaného výše, ale má tu výhodu, že bude sestaven automaticky z vlastností, které určíme pro jednotlivé položky. V případě aplikace to může být Název a Prostředí (měli bychom rozlišit tu samou aplikaci, ale instalovanou v Produkci nebo Testovacím prostředí), v případě databáze Databázový server a Název databáze, v případě hardware pak již zmiňovaný Výrobce, Typ/Model a Sériové číslo. Jde o identifikující vlastnosti. Pokud budeme chtít z popisu jednoznačně poznat, o jakou položku se jedná, právě tyto vlastnosti nám to umožní. Oproti výše zmíněnému konceptu unikátního a neměnného názvu je tu ještě jeden rozdíl – změna popisu v času.

Popis se může v čase měnit a je to v pořádku

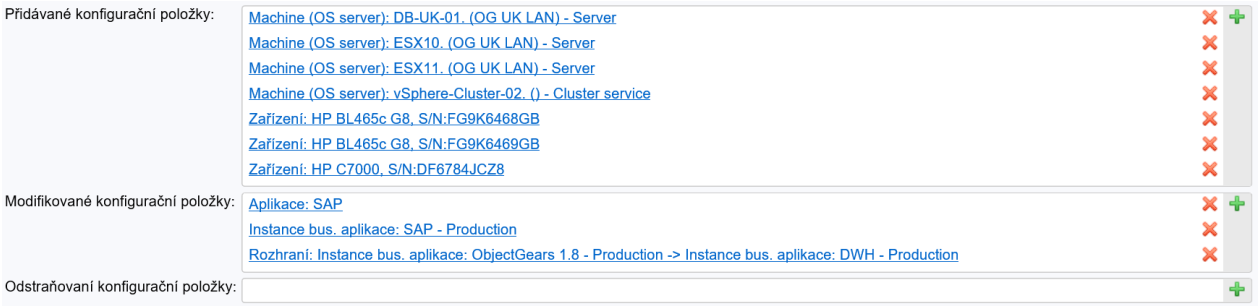
Představte si, že v rámci plánované změny přesouváte databázi z jednoho serveru na jiný. Bude to stále ta samá databáze (ta samá konfigurační položka), ale mění se jedna z jejích identifikačních vlastností (databázový server). V rámci změny upravíte connection string aplikací, které s databází pracují, a změníte vlastnost Databázový server u dané položky v konfigurační databázi. Podobně bychom k věci přistoupili, pokud bychom měnili název databáze. Nevznikne nám při tom nová konfigurační položka a potřeba změny názvu databáze skutečně může nastat. Automaticky se nám pak změní popisek, podle kterého položku vyhledáváme. Pokud bychom se drželi konceptu neměnného názvu položku, nebylo by možné tuto změnu provést.

Typ položky součástí jejího popisu

Dále bude vhodné do popisu položky zahrnout i typ položky. Pokud budeme vybírat ze seznamu konfiguračních položek, hned nám bude jasné, o jaké položky jde.

Na obrázku níže vidíme příklad z Projektu. První přidávanou položkou je např. server DB-UK-01 nacházející se v síti OG UK LAN. V modifikovaných položkách vidíme aplikaci SAP, pak její konkrétní

instanci používanou v Produkci a pak Rozhraní (interface) mezi aplikací produkční instancí ObjectGears a produkční instancí Datového skladu (DWH).



Obrázek 7: Část karty projektu mapující dopad projektu na konfigurační položky.

Konkrétní aspekty hlavních typů konfiguračních položek

Služby

Služby představují zajištění konkrétní potřeby businessu. Rozlišujeme business služby a technické služby. Příkladem business služeb může být:

- Zavést nového business partnera/zákazníka/dodavatele
- Vystavit objednávku
- Zkontrolovat bonitu partnera
- Vyfakturovat zákazníkovi naši dodávku
- Zaúčtovat fakturu
- Zaplatit fakturu

Business služby jsou zajištěny technickými službami (viz pojetí business a technických služeb v ITIL). V praxi doporučujeme umožnit vazbu business služeb také přímo na další konfigurační položky. Ze zkušenosti víme, že při implementaci je vůle vyjádřit to, že business služba je zajišťována určitou aplikací, a vytváření technické služby mezi business službou a aplikací přijde lidem často zbytečně komplikované.

Aplikace

V praxi se často setkáme s tím, že určitá aplikace je příliš velkým celkem pro vyjádření toho, co je v aplikaci dotčeno. Např. firma používající SAP, může používat jednu instanci této aplikace napříč organizací. V této situaci je vhodné rozlišovat i jednotlivé moduly či funkční celky. Kromě konfigurační položky SAP, tak vytvoříme i další, u kterých ve vlastnosti „Je modulem“ vybereme SAP. Takto můžeme pracovat s konfiguračními položkami, které nazveme např. SAP FI, SAP MM atd. Podobný přístup se nám bude hodit u mnohých ERP systémů. Pokud budeme řešit incident, můžeme zvolením konkrétního modulu lépe lokalizovat část systému, která je postižena. Každý modul může mít také rozdílné business vlastníky.

Zatímco u jiných entit (např. databáze, zařízení-hardware, server…) zpravidla panuje jasnost v tom, co danou entitou je a co není, u aplikací tomu tak nemusí být. V takových případech doporučujeme vzít v úvahu následující kritéria:

- Business daný systém považuje za důležitou aplikaci, kterou potřebuje ve svých procesech
- V případě výpadku je pravděpodobné, že bude kontaktován Service desk
- Dodává výstupy, s nimiž business dále pracuje
- Vyšší počet uživatelů
- Vyžaduje licencování

Kromě vlastností aplikací se budeme zabývat také jejich návaznostmi na další entity. Za výchozí stav doporučujeme vzít model dodávaný s ObjectGears a upravit ho dle vašich potřeb. U aplikací bychom měli zvážit následující vazby s dalšími entitami.

- Rozhraní (interfacy) – tato entita je definována aplikací předávající data a aplikací přijímající data.
- Datové zdroje – databáze nebo jiné datové zdroje používané aplikací
- Infrastruktura – propojení je realizováno přes aplikační komponentu, která běží na určitém serveru
- Procesy – aplikace podporuje určitý proces
- Klíčoví uživatelé (Pracovní pozice)
- Entity Change a Release managementu, Test a Project managementu, Incident and Problem managementu, Business a Technických služeb, které s aplikacemi pracují jako s ostatními konfiguračními položkami

Rozhraní (Interfacy)

Interfacy představují zásadní vztah mezi dvěma konfiguračními položkami – aplikacemi. Evidujeme aplikaci poskytující data a aplikací přijímající data. Zajímat nás bude kromě popisu důvodu komunikace, frekvence přenosu dat, přenášená datová entita, konkrétní rozhraní a technologie rozhraní.

Datové zdroje

Data v databázích mohou být sdílena různými aplikacemi. Tyto závislosti je vhodné sledovat. U databází nás zajímají informace jako název, na jakém databázovém serveru se nachází, kolace, úroveň kompatibility, datum poslední zálohy, model obnovy, velikost, dostupné místo. Kromě klasických databází nacházejících se na databázových serverech se můžeme setkat i s dalšími zdroji dat. Business někdy používá tzv. **End User Computing** aplikace. Jde například o aplikace založené na MS Access nebo MS Excel. Setkat se můžeme i s jinými aplikacemi pracujícími s vlastními datovými základnami. V případě, že tyto aplikace podporují kritické procesy, měli bychom v konfigurační databázi podchytit i jejich datové zdroje. Pokud takto zachytíme skutečný stav věcí v naší organizaci, učiníme první krok ke zlepšení v oblasti závislosti na neadekvátních aplikací. Zmapování a pojmenování je prvním krokem k nápravě.

Servery

Je vhodné rozlišovat servery jakožto systémy s operačním systémem a zařízeními (hardware). Zařízení jsou fyzické krabice instalované do racků nebo nacházející se v budovách mimo racky. Na zařízení se instalují servery. Klíčovou vlastností serveru je operační systém, např. Windows Server 2016, Windows Server 2019, Linux atd. Obě entity mají rozdílné vlastnosti.

Servery mohou být **fyzické (nativní)**, které jsou instalované na hardware, a **virtuální**, které běží na virtualizační software (např. VMware nebo Hyper-V). Dále bychom měli zachytit **clustery**, sestávající

z nódů fyzických serverů, **clusterové služby** běžící na clusterech či **farmy** sdružující několik fyzických či virtuálních serverů. U serverů budeme pravděpodobně sledovat mnoho dalších vlastností – DNS jméno, zda používá DHCP rezervaci, IP adresu (v primární a sekundární lokalitě), okno údržby, název serveru, kategorie patchování, do níž patří, bezpečnostní zóna, v níž se nachází, počet CPU, velikost RAM, velikost disků, Windows doména, preferovaný, aktuální a failover nód …

Zařízení (Hardware)

U zařízení nás zajímá výrobce, typ/model, sériové číslo, kde je zařízení fyzicky umístěno, v jaké bezpečnostní zóně se nachází, počet CPU, jejich typ a frekvence, velikost RAM, velikost fyzických disků, vlastník, inventární číslo, rodičovské zařízení (např. blade enclosure pro blade servery), rok výroby, konec podpory, servisní kontrakt…

Automatická vs. manuální aktualizace data

Mnohé vlastnosti není třeba vyplňovat ručně. Můžeme je vyčítat ze systémů pomocí skenů infrastruktury. Takovéto vlastnosti bývá vhodné nastavit pouze ke čtení, protože se aktualizují sami. Příkladem těchto dat je hardware, servery, databáze či certifikáty. Automaticky můžeme zjišťovat i software instalovaný na počítačích. Většinu business aplikací, které používáme, již zpravidla není možné nechat jen naskenovat. Může jít o moduly větších systémů nebo naopak drobné aplikace, které nejsou nainstalované. Podobné je to s rozhraními/interfacy. Je možné analyzovat komunikaci aplikací mezi sebou, ale jasnější informace obvykle získáme při rozhovorech se správci aplikací.

Neváhejte jako zdroje dat použít systémy třetích stran. Data z různých systémů sice často umí vyčítat přímo software konfigurační databáze (viz např. infrastrukturní řešení ObjectGears), ale tato schopnost by neměla být hlavním kritériem při rozhodování se o konkrétním řešení pro konfigurační databázi.
V dnešní době existuje řada specializovaných nástrojů, které získají data z nejrůznějších systémů a uloží je ve formě souborů, které pak lze do konfigurační databáze načíst. Mnohé systémy dnes lze ovládat (a vyčítat z nich data) pomocí **PowerShell**. Nejedná se jen systémy Microsoftu, ale i jiných dodavatelů, kteří implementují **MMC (Microsoft Management Console) snap-ins**. Mnohem důležitější jsou u konfigurační databáze schopnosti data modelovat, vizualizovat a používat v dalších procesech.

Baseline

V určitých případech budeme mít zájem sledovat, zda položky vykazují určité (schválené) vlastnosti nebo zda došlo u těchto vlastností k odchylce. U každé položky takto můžete sledovat kritická data a po oskenování skutečného stavu porovnat schválené a skutečné hodnoty. Na rozdíly můžete být upozorněni při zobrazení dané položky, automatickou emailovou notifikací, nebo když si otevřeme příslušný report. Baseline vždy můžete jednoduše upravit, pokud chcete prohlásit skutečný stav za schválený a na odchylky již nebýt upozorňováni.

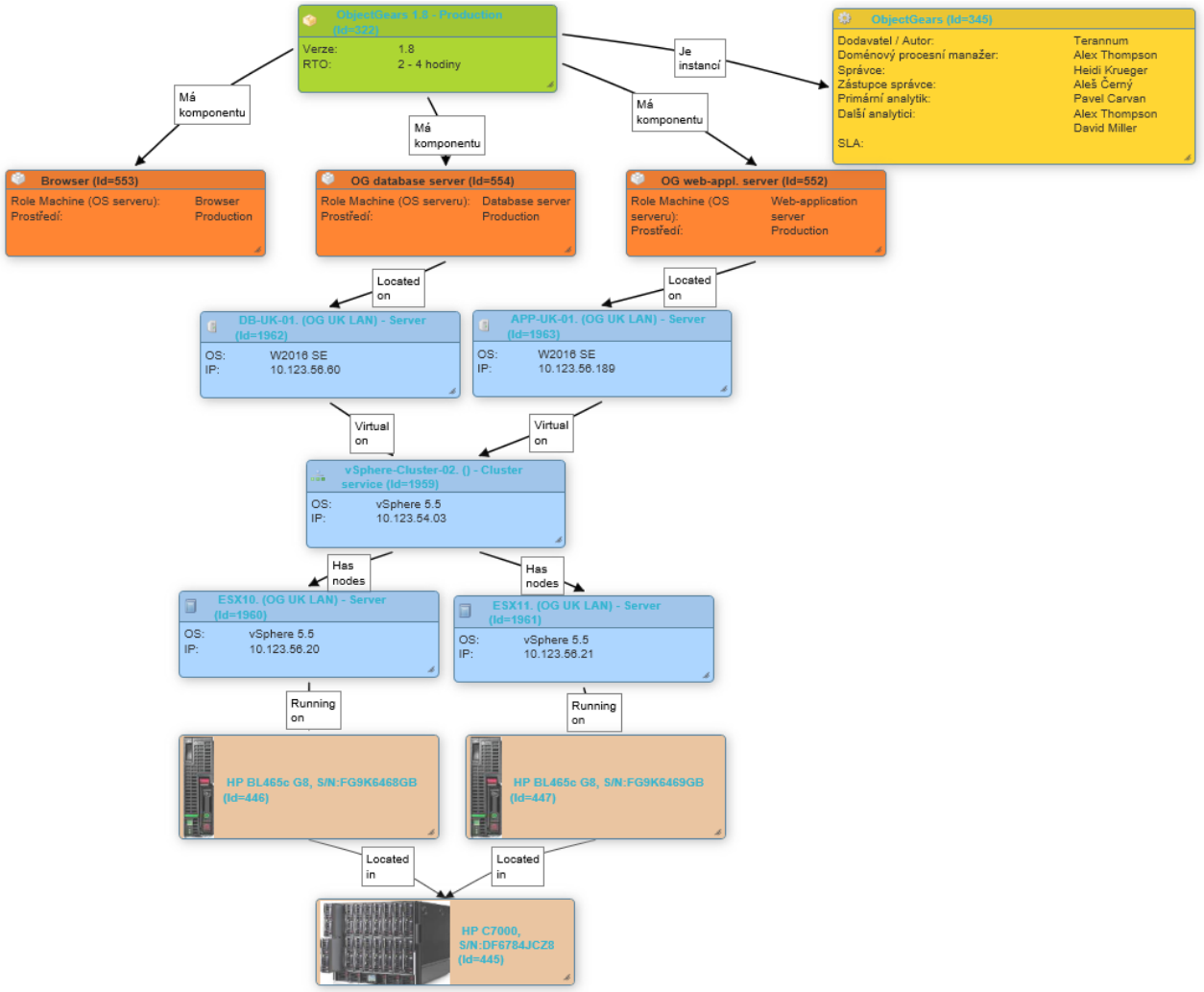
Propagace manuálně nastavených dat do systémů

Data v konfigurační databázi můžete nejen sbírat z cílových systémů, ale také můžete naopak cílové systémy nastavovat podle hodnot zadaných do konfigurační databáze. Moderní systémy lze spravovat

nejen přes grafické uživatelské rozhraní, ale také přes jejich programová rozhraní. Na tlačítko tak můžete například vytvořit virtuální servery podle parametrů, které jste zvolili v Konfigurační databázi.

Vizualizace dat

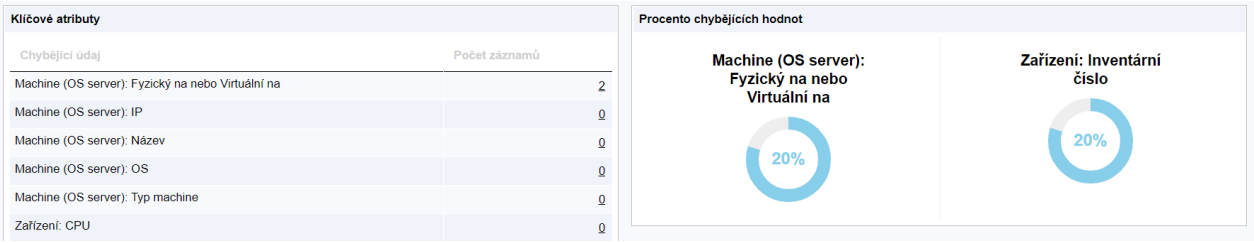
Data v konfigurační databázi bychom měli být schopni vizualizovat. Schémata umožňují soustředit pozornost na konkrétní vztahy a pochopit rychle potřebné souvislosti.



Obrázek 8: Schéma znázorňující vztahy konfiguračních položek.

Chybějící kritická data

U každé entity bychom si měli stanovit, které vlastnosti jsou zásadní (povinné), a míru jejich vyplnění bychom měli pravidelně sledovat. Chybějící data by měl zajistit vlastník dané konfigurační položky.



Obrázek 9: Chybějící data v klíčových vlastnostech – počet a jejich relativní podíl, drill-down na dané záznamy.

Shrnutí

Tento průvodce poskytuje základní rámec pro design konfigurační databáze. Zde je shrnutí základních konceptů:

- Konfigurační databáze modeluje služby, aplikace a infrastrukturu, na níž jsou založeny (datové zdroje, servery, hardware atd.).
- Úspěch implementace konfigurační databáze a procesů řízení konfigurace závisí na schopnosti demonstrovat přidanou hodnotu.
- Při implementaci je třeba reflektovat kulturní aspekty, aby byl tento systém pozitivně přijat.
- Implementaci a rozvoj je dobré plánovat. Spoléhat se na organický rozvoj bez pravidelné revize plánu nevede k optimálním výsledků.
- Vytvořte řídící výbor, který bude pravidelně řešit vzniklé otázky a problémy a prioritizovat rozvoj konfigurační databáze.
- Upravte výchozí model Konfigurační databáze dle vašich potřeb.
- Definujte *Krátký popis* každé třídy konfiguračních položek jako kombinaci identifikujících vlastností těchto položek. Tento popis bude dynamický a bude reflektovat změnu identifikujících vlastností. Jako neměnný identifikátor používejte sekvenci generovanou automaticky konfigurační databází.
- Věnujte pozornost tomu, co budete považovat za aplikaci. Určité konfigurační položky mají zcela jasnou definici toho, co je za ně považováno a co ne. U aplikací zpravidla nejde o triviální definicí a bude třeba brát v úvahu více aspektů.
- Definujte vlastnosti a vazby mezi konfiguračními položkami. Kritické vlastnosti a vazby sledujte v přehledech chybějících dat. Domluvte se na zodpovědnostech: kdo, dokdy a jaká data doplní. Vyčítejte data z infrastruktury automatickými skeny a případně, kde je to vhodné, porovnávejte skutečný stav se zamýšleným/schváleným.
- Na konfigurační databázi navažte další procesy (Incident a Problem management, Change a Release management, Project management). Vyjasníte tím závislosti a při řešení úkolů nebo plánování budete znát historii a kontext položky.

Kontakt:

Pavel Carvan Autor se dlouhodobě věnuje problematice Configuration management.	Email: pavel.carvan@objectgears.cz LinkedIn: https://www.linkedin.com/in/pavel-carvan-96676835/