

Školení

ArchiMate 3.1 – foundation

Verze 1.11

Příprava na certifikaci



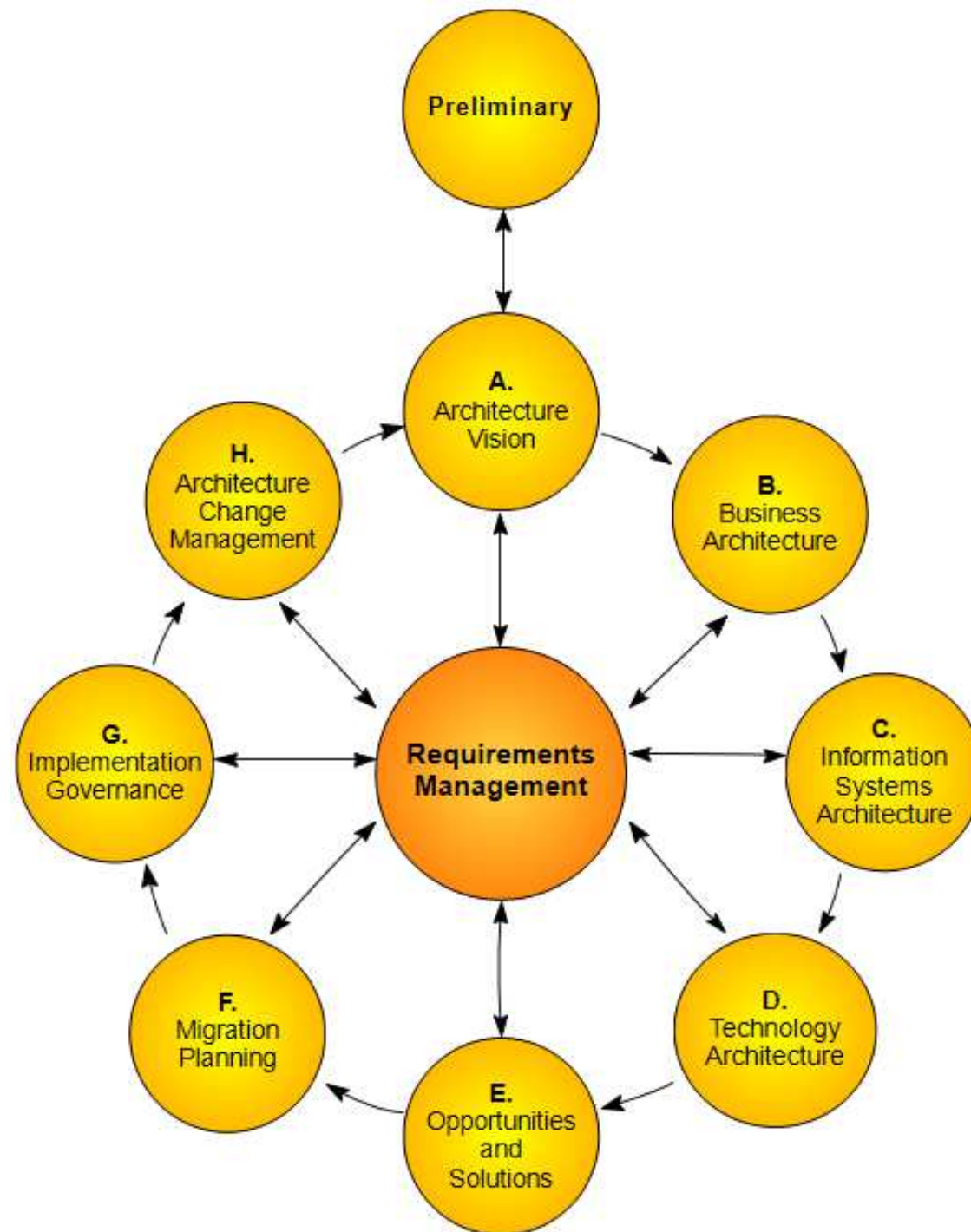
ArchMate

základní informace

Co je TOGAF

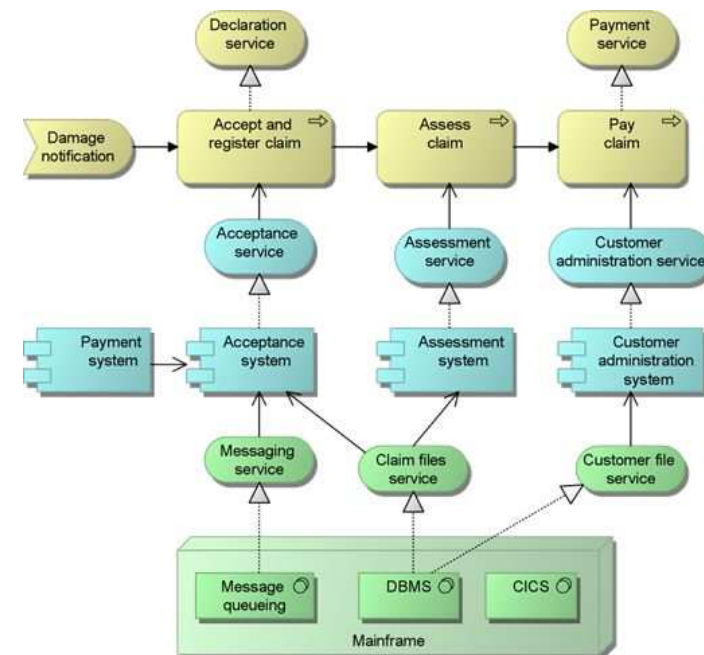
(The Open Group Architecture Framework)

- TOGAF je architektonický framework, konkrétně The Open Group Architecture Framework.
- Poskytuje metodiku a prostředky pro podporu přípravy, využívání, rozvoje, změny a údržby podnikové architektury.
- Je založen na iterativním přístupu k rozvoji podnikové architektury (a jejího modelu).
- Je aplikovatelný na rozsáhlé projekty (programy) i na malé, jednoduché změny.
- Vyžaduje customizaci.
- Sestává z osmi základních fází + dvě doplňkové.
- Definuje postupy, úložiště modelů, výstupy jednotlivých fází, základní principy řízení změn atd.

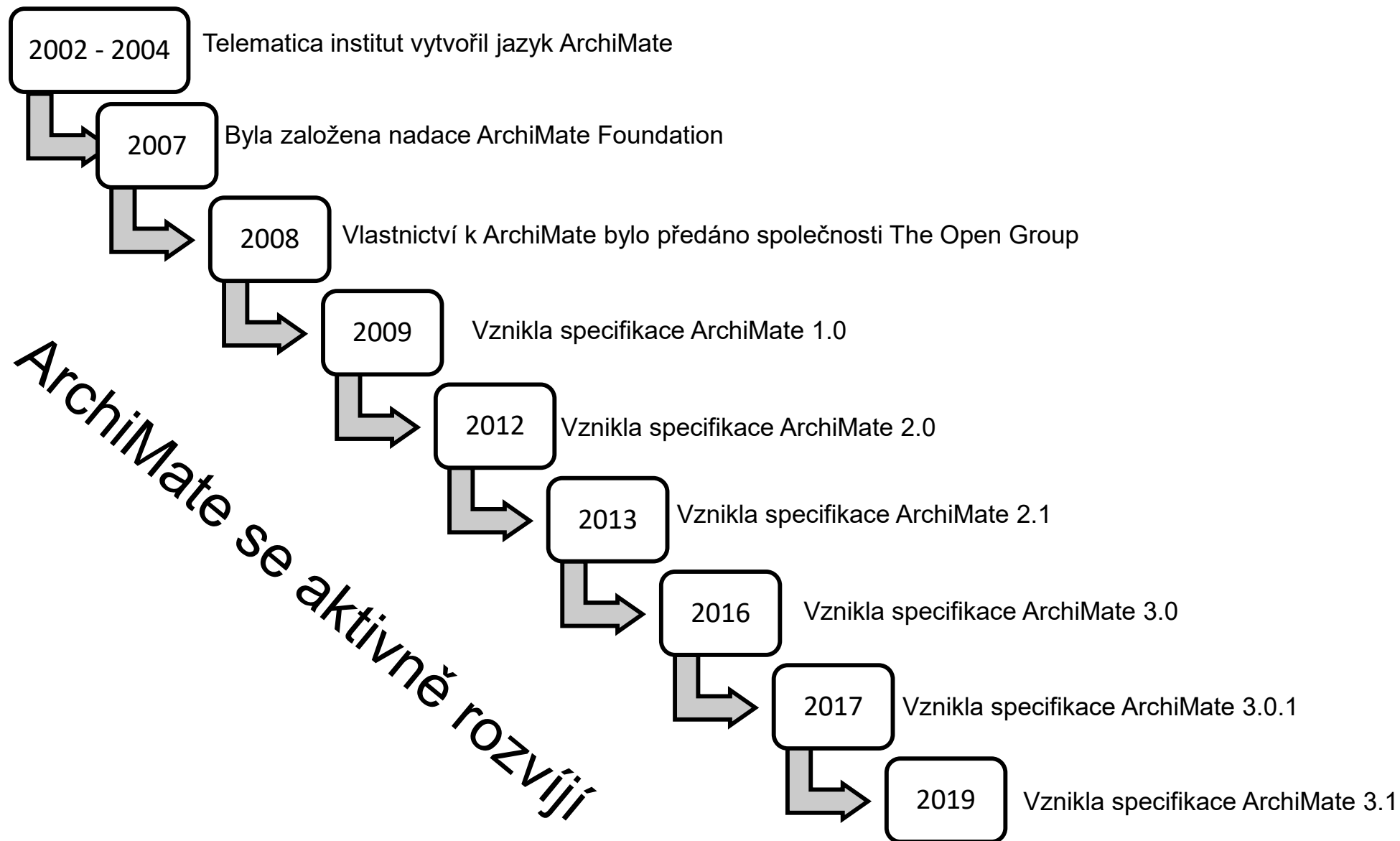


Co je ArchiMate

- Jedná se o modelovací jazyk, který slouží pro popis, analýzu a vizualizaci podnikové architektury.
- Jedná se o standardizovaný jazyk, který velmi konkrétně definuje, jak zakreslovat různé pohledy na podnikovou architekturu.
- Definuje svůj vlastní set elementů a vazeb a určuje jejich význam.
- Některé principy jsou odvozeny z jazyka UML.
- Vývoj ArchiMate je velmi úzce svázán s architektonickým frameworkem TOGAF a to včetně sdílené terminologie.

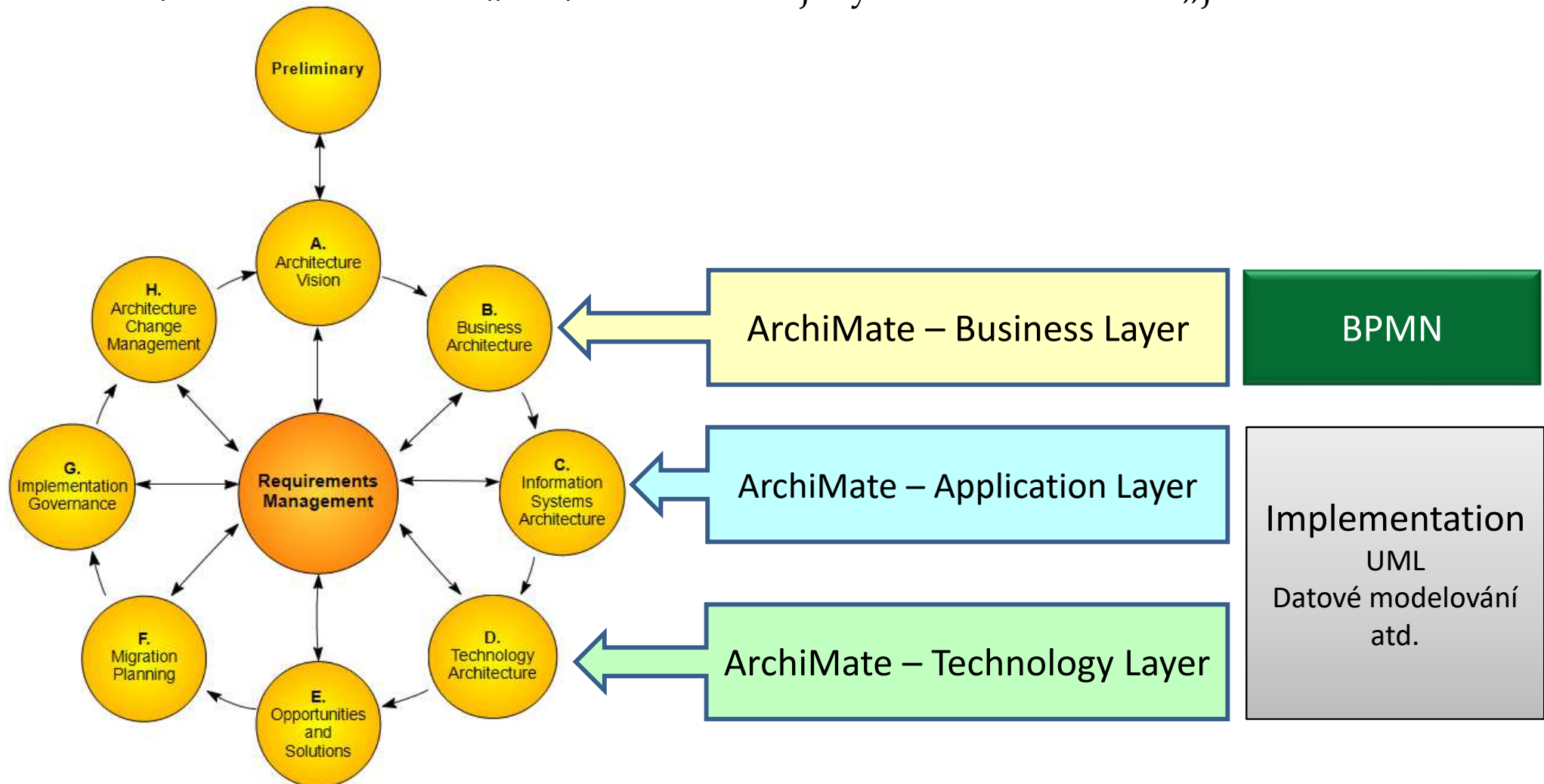


Evoluce ArchiMate



Vztah TOGAF a ArchiMate

Jazyk ArchiMate doplňuje a rozvíjí TOGAF. Zjednodušeně se dá říci, že v některých místech, kde TOGAF říká „co“, modelovací jazyk ArchiMate říká „jak“.



Proč používat ArchiMate



- Společný jazyk pro architekty – pokud někdo zná jazyk ArchiMate, porozumí vašim modelům a tomu, co jste ve svých modelech zachytili.
- Jeden obrázek za tisíc slov – lidský mozek snadněji přijímá rozsáhlé a komplexní informace v grafické formě než ve formě zhuštěného textu.
- Lepší komunikace uvnitř společnosti – ArchiMate umožňuje businessu a IT komunikovat společným jazykem, používat společnou terminologii a jiné výrazové prostředky. Navíc, čtení diagramů zakreslených v ArchiMate není složité a snadno pochopitelné i pro lidi mimo IT.
- Odstranění nejednoznačností – jazyk ArchiMate má jasně danou strukturu a pravidla. Oproti psanému textu (a i oproti některým jiným modelovacím jazykům) je zakreslený diagram jednoznačný a neumožňuje dvojí význam.
- Umožňuje dopadové analýzy – oproti psanému textu je lehké zjistit závislosti například business procesů na aplikacích a třeba aplikací na hardware.
- Kratší zaškolování – standardizovaný jazyk znamená menší potřebu zaškolení nových zaměstnanců a rovněž je mnohem větší šance, že se podaří získat nové zaměstnance, kteří již znalostí tohoto modelovacího jazyka disponují.
- Přímá vazba na TOGAF – pokud společnost používá TOGAF jako architektonický framework, je ArchiMate jeho vhodným doplněním.

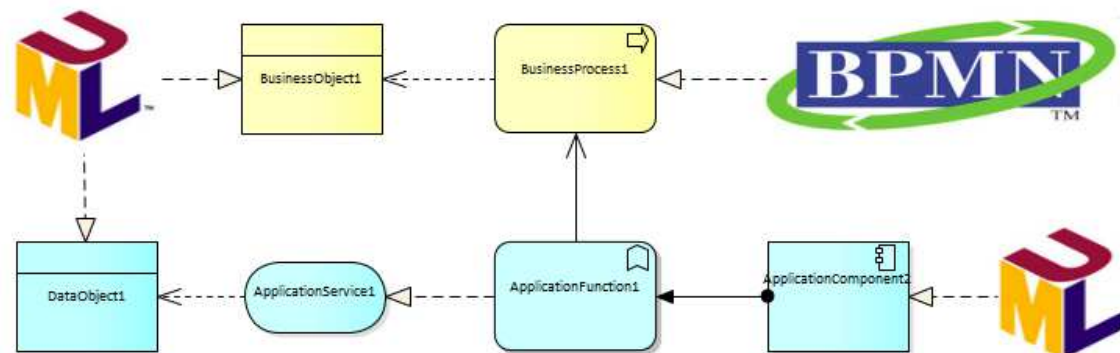


Omezení ArchiMate

- ArchiMate též není všemocný. Jeho slabinou (ale i výhodou) je to, že se zaměřuje pouze na architekturu (na druhou stranu to však dělá dobře).
- Pomocí samotného ArchiMate však nelze popsat vše potřebné – pokud není možné omezit se skutečně pouze na architekturu - a je nutné jej kombinovat s jinými jazyky.
- V takovém případě je vhodné zvolit pro zakreslování modelů takový nástroj, který takové kombinování syntaxí umožní (například **Enterprise Architect**, který tuto problematiku zvládá velmi dobře).

Některá omezení:

- Zakreslení procesů – ArchiMate nabízí jen základní prvky pro zakreslení business procesů, pokud je potřebné tyto procesy zakreslit do většího detailu, je vhodné použít například BPMN nebo Act. Diag.
- Popis aplikačních komponent – pokud je potřeba popsat vnitřní strukturu aplikací či informačních systémů, je vhodné využít například notaci UML, která k tomu poskytuje potřebné prostředky.
- Business objekty – ArchiMate nedovoluje přidávat atributy (podobně jako v třídních diagramech).



Open Group != OMG (Object Management Group)

The Open Group je nezávislá soukromá společnost, která vlastní ochranné známky například na:

- TOGAF
- ArchiMate

Nejedná se v principu o otevřené sdružení, byť se tak jmenuje. Rovněž používání jejich standardů v některých případech podléhá licencování.



Object Management Group je otevřené sdružení s jasnými pravidly a zajišťuje zejména standardizaci syntaxí, jako jsou například:

- UML
- BPMN
- SysML
- CORBA

...a mnoho dalšího. Jejich standardy zpravidla licencování nepodléhají (zejména ty hlavní, UML, BPMN apod.)



ArchiMate

základní principy

Vrstvy (layers) a aspekty (aspects)

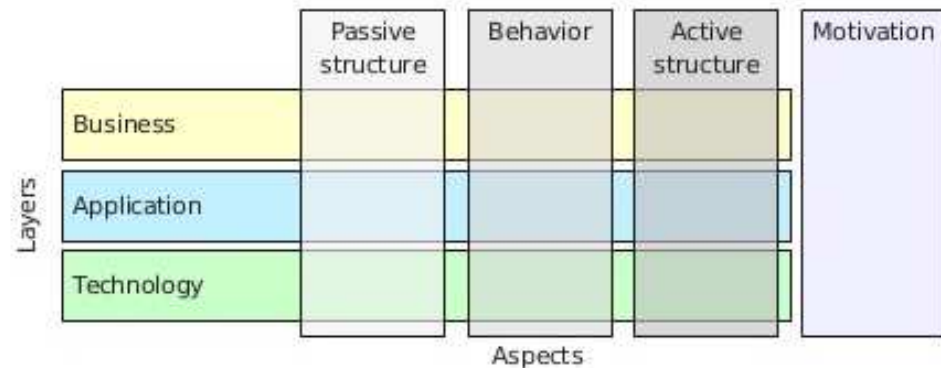
ArchiMate je založen na dvou základních principech:

- Vrstvy – ArchiMate definuje tři základní vrstvy (Business, Application a Technology), které se zaměřují na konkrétní části architektury.
- Aspekty – ArchiMate striktně odděluje tři základní aspekty:
 - Aktivní prvky
 - Chování aktivních prvků
 - Pasivní prvky

Tento základ se nazývá ArchiMate core.

Dále obsahuje několik extenzí (z nichž některé přidávají další vrstvy):

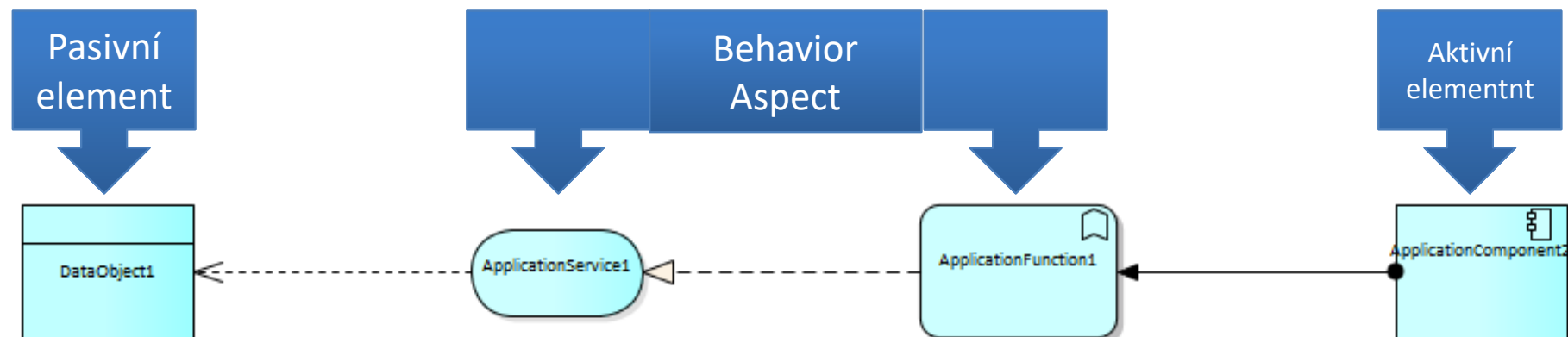
- Physical (ArchiMate 3.0)
- Implementation & Migration
- Motivation
- Strategy



Aspekty (aspects)

ArchiMate striktně odděluje tři základní aspekty:

- Active Structure Aspect – reprezentují strukturální elementy (např. aktory, role, komponenty, hardware), které jsou schopny určitého chování. Také jsou tyto elementy označovány jako „aktivní“. Elementům se někdy říká také prvky.
- Behavior Aspect – reprezentuje chování (např. procesy, funkce, služby) prováděné aktivními (strukturálními) elementy. Chování je přiřazeno (assigned) aktivnímu elementu.
- Passive Structure Aspect (pasivní elementy) – reprezentují objekty (např. business objekty, datové objekty, ale i fyzické objekty), na které může být aplikováno chování aktivního elementu (či jsou takovým chováním ovlivněny).

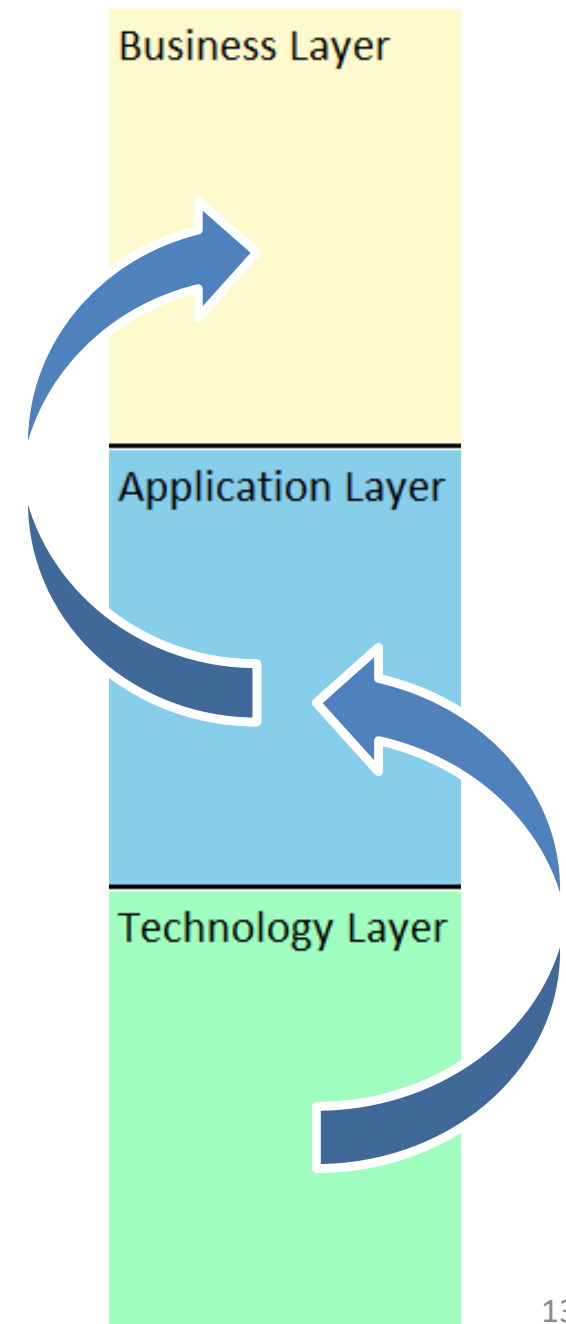


Vrstvy (layers) – Archimate Core

Základ ArchiMate se zabývá třemi základními vrstvami:

- Business Layer – v této vrstvě jsou předmětem zájmu například produkty, procesy, role, aktéři, business objekty, kontrakty.
- Application Layer – zabývá se datovou a aplikační doménou - například aplikační komponenty, aplikační funkce, aplikační služby, datové objekty. Tato vrstva primárně podporuje business vrstvu (Business Layer).
- Technology Layer – zabývá se technickými prvky - například servery, systémovým softwarem, sítěmi, datovými toky a obecně hardwarem. Příkladem může být node, device, communication network. Tato vrstva primárně podporuje aplikační vrstvu (Application Layer)

Základní myšlenka je, že nižší vrstva podporuje svými elementy a chováním vyšší vrstvu.





Vrstvy (layers) – ArchiMate Extensions

Rozšíření jazyka ArchiMate doplňuje tři základní vrstvy o tři další:

- Strategy Layer – poskytuje prostředky pro strategické plánování a definuje čtyři elementy – zdroje (Resource), schopnosti (Capability), plán / přístup (Course of action) a Value Stream.
- Physical Layer – definuje elementy potřebné pro modelování fyzické reality, například materiál (Material), zařízení (Facility) nebo distribuční síť (Distribution Network). Pomůcka pro odlišení Technologické vrstvy of Fyzické: pokud má zařízení nebo nástroj (Equipment) schopnost ovlivňovat fyzickou realitu – něco vyrobit, něco někam přenést apod. jedná se o fyzickou vrstvu. Pokud se aktivita odehrává pouze na úrovni software, či uvnitř HW, jedná se o technologickou vrstvu. Tato vrstva nedefinuje samostatné chování (behavior), to využívá z Technology Layer.
- Implementation & Migration Layer – definuje elementy potřebné pro plánování a migraci projektů (například Work package, Delivery. Gap). Pozor, nejedná se o implementaci systémů (to ArchiMate neumí), ale o implementaci projektů.
- Motivation Extension – doplňuje ArchiMate o elementy, které umožňují zaznamenat doplňující informace typu Stakeholder, Requirement, Principle, Driver, Constraint a další a rovněž jak se navzájem ovlivňují.

Strategy Layer

Business Layer

Application Layer

Technology Layer

Physical Layer

Implementation & Migration Layer

Vrstvy a Aspekty – kompletní přehled

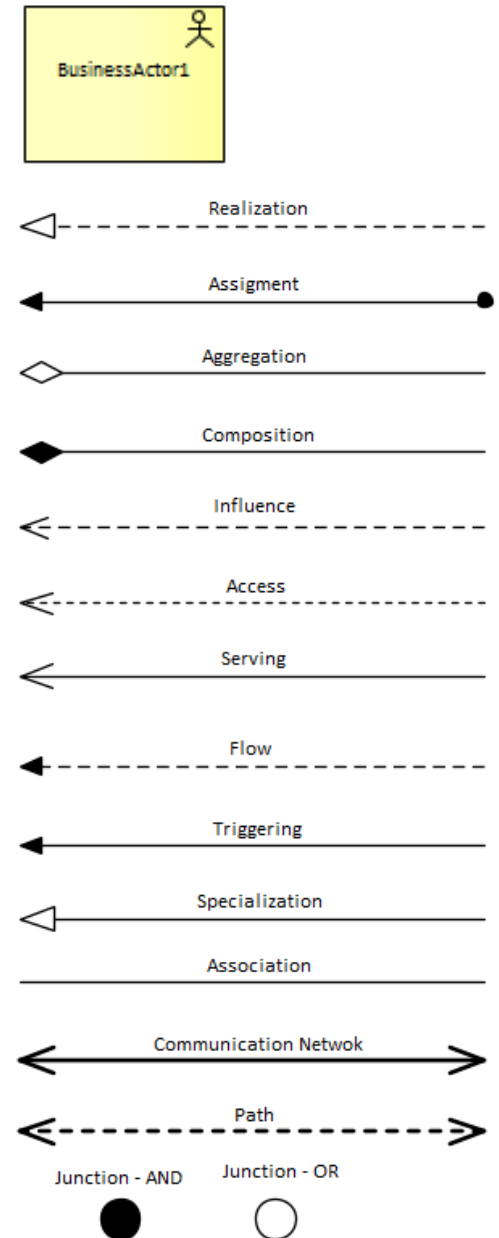
	Passive structure		Behavior		Active Structure		Motivation
Strategy Layer							
Business Layer							
Application Layer							
Technology Layer							
Physical Layer							
Implementation							

ArchiMate

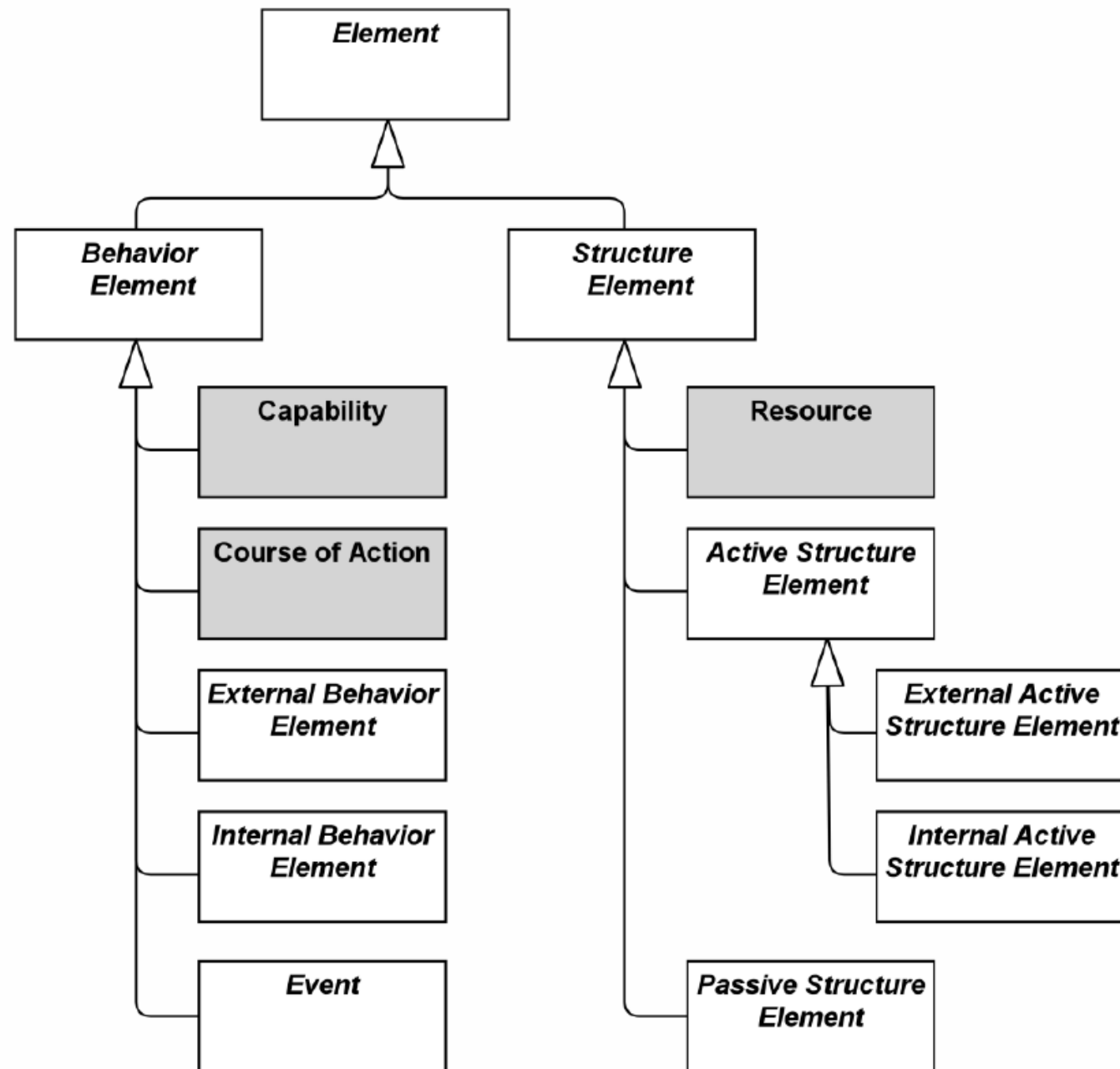
základní prvky

Základní prvky ArchiMate

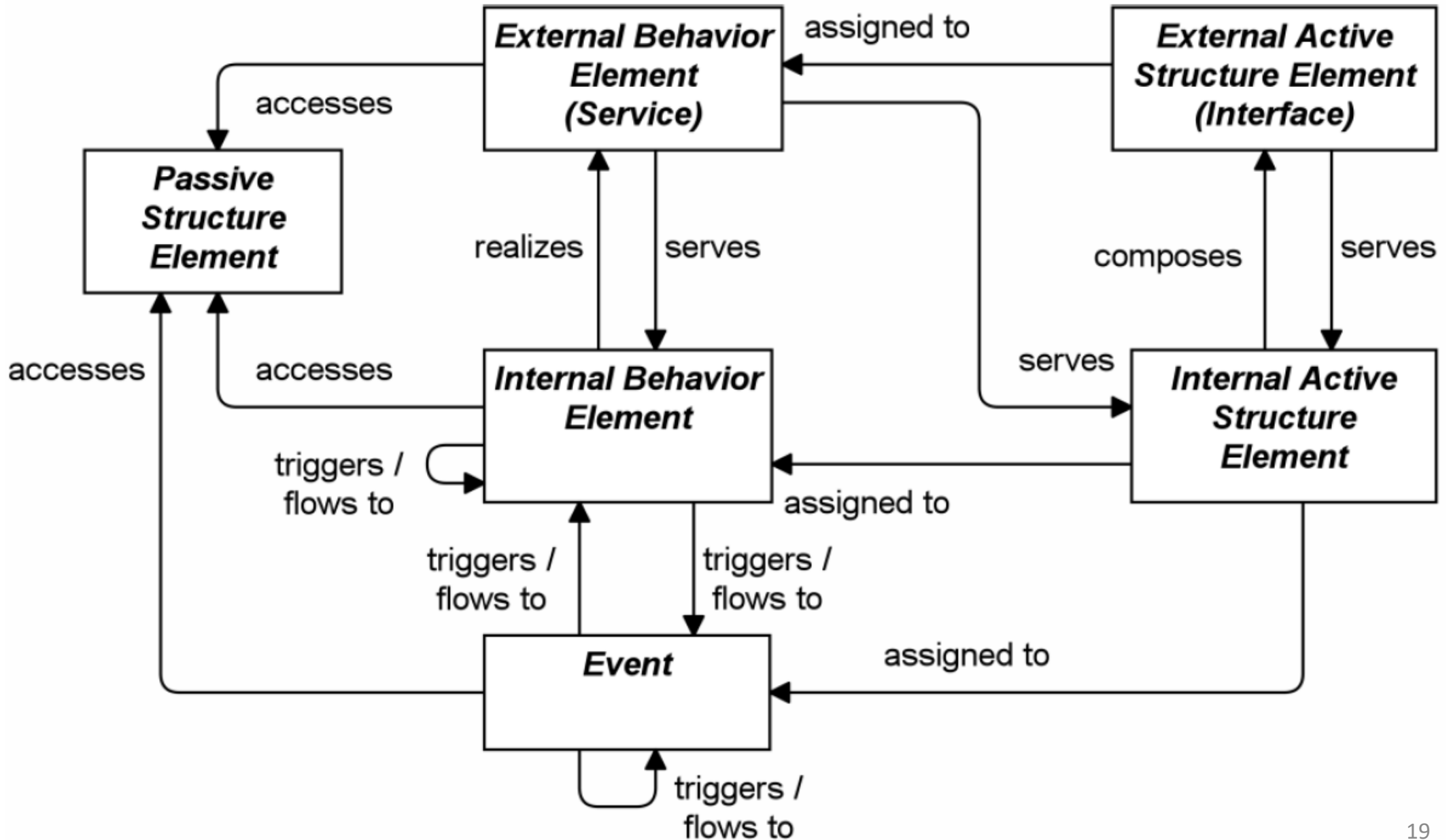
- Elementy (Elements) – základní prvek archimate. Popisuje strukturu nebo chování. Pomůcka: u elementů je dobré sledovat tvar rohů a barvu, což usnadňuje jejich identifikaci a zařazení – viz kapitola elementy.
- Kompozitní elementy (Composite Elements) – jedná se o elementy složené z více elementů (lze i z různých vrstev či aspektů).
- Vazby (Relationships) – definují vztah mezi elementy. Každá vazba má svůj význam. Vazby se dělí do skupin (Structural, Dependency, Dynamic, Others) a rovněž jsou řazeny dle své síly – například Aggregation je slabší než Composition.
- Atributy (Attribute) – vlastnosti náležející elementům či vazbám.
- Koncept (Concept) – obecný název pro element či vazbu.
- Vrstva (Layer) – úroveň abstrakce definovaná jazykem ArchiMate, která může být modelována.
- Aspekt (Aspect) – charakteristika elementu (která není závislá na vrstvě – proto vrstvy a aspekty tvoří matici) související se zájmy příslušných stakeholderů. Používá se rovněž pro definici struktury jazyka ArchiMate.
- Model - kolekce konceptů v kontextu jazyka ArchiMate.
- Core Elements – elementy ze tří základních vrstev ArchiMate (Business Layer, Application Layer a Technology Layer).
- Viewpoint – konkrétní pohled na model (realizovaný formou diagramu) z hlediska určitého stakeholdera.



Archimate – metamodel #1



Archimate – metamodel #2

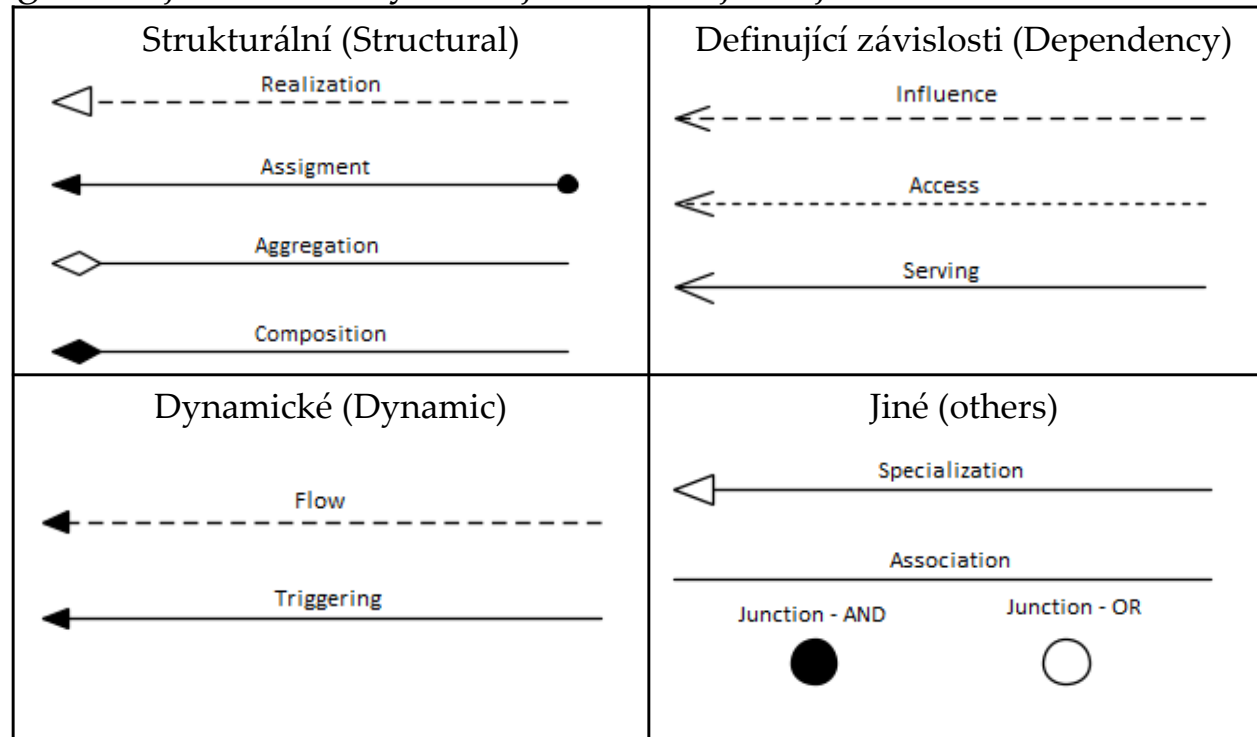


ArchiMate

Vazby (Relationships)

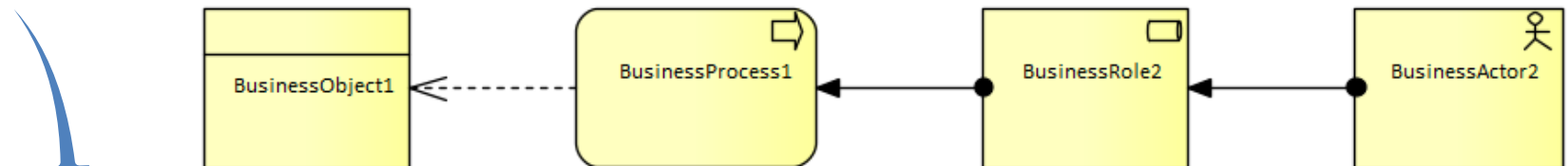
Vazby (relationships) - přehled

Vazby se v jazyce ArchiMate dělí na čtyři skupiny (mimo specifických vazeb v Technology Layer). Vazby, v jednotlivých kategoriích, jsou seřazeny od nejslabší k nejsilnější:



Vazby mohou být také seřazeny dle své síly (od nejslabšího k nejsilnějšímu):

1. Influence – nejslabší
2. Access
3. Serving
4. Realization
5. Assignment
6. Aggregation
7. Composition - nejsilnější

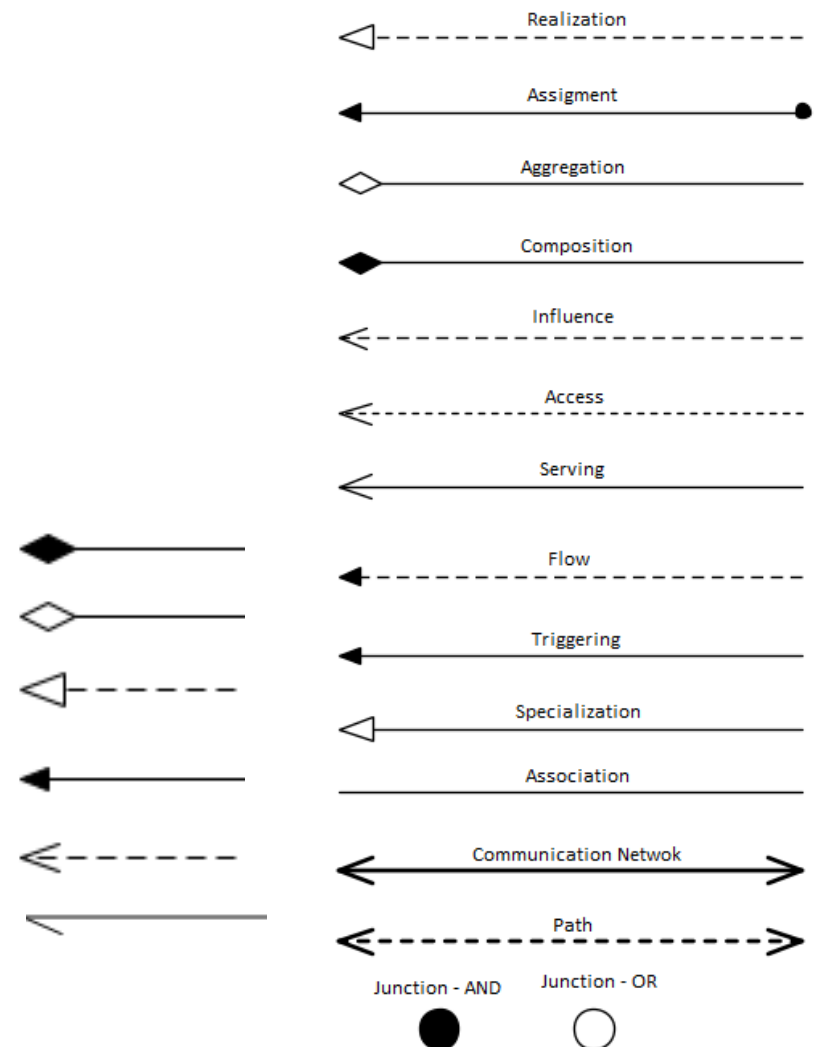


Rozpoznávací znaky vazeb

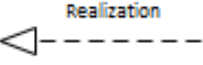
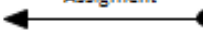
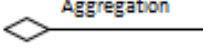
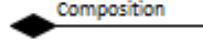
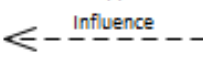
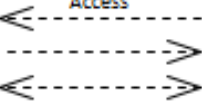
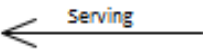
U vazeb rozpoznáváme následující znaky, podle kterých dokážeme příslušnou vazbu identifikovat (další možností je samozřejmě kontext – mezi jakými objekty je vazba natažena):

- Počátek vazby
 - Prostá čára dle typu vazby ——— ————
 - Vyplněný kruh (či spíše kulička) ———●
- Tělo vazby
 - Plná čára ———
 - Čerchovaná - - - - -
 - Drobně čerchovaná
- Šipka
 - Diamond vyplněný
 - Diamond prázdný
 - Šipka uzavřená, plná čára šipky, nevyplněný vnitřek
 - Šipka uzavřená, plná čára šipky, vyplněný vnitřek
 - Šipka otevřená, plná čára šipky
 - Šipka otevřená, čerchovaná čára šipky

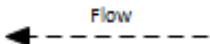
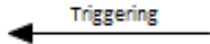
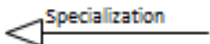
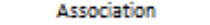
A kombinace toho všeho...



Vazby - definice #1

Vazba	Definice	Typ
 Realization	The realization relationship indicates that an entity plays a critical role in the creation, achievement, sustenance, or operation of a more abstract entity.	Structural
 Assignment	The assignment relationship expresses the allocation of responsibility, performance of behavior, or execution.	Structural
 Aggregation	The aggregation relationship indicates that an element groups a number of other elements.	Structural
 Composition	The composition relationship indicates that an element consists of one or more other elements.	Structural
 ++ Influence	The influence relationship models that an element affects the implementation or achievement of some motivation element.	Dependency
 Access	The access relationship models the ability of behavior and active structure elements to observe or act upon passive structure elements.	Dependency
 Serving	The serving relationship models that an element provides its functionality to another element.	Dependency

Vazby - definice #2

Vazba	Definice	Typ
	The flow relationship represents transfer from one element to another.	Dynamic
	The triggering relationship describes a temporal or causal relationship between elements.	Dynamic
	The specialization relationship indicates that an element is a particular kind of another element.	Other
	An association models an unspecified relationship, or one that is not represented by another ArchiMate relationship.	Other
	A junction is used to connect relationships of the same type.	Other

Seznam povolených vazeb

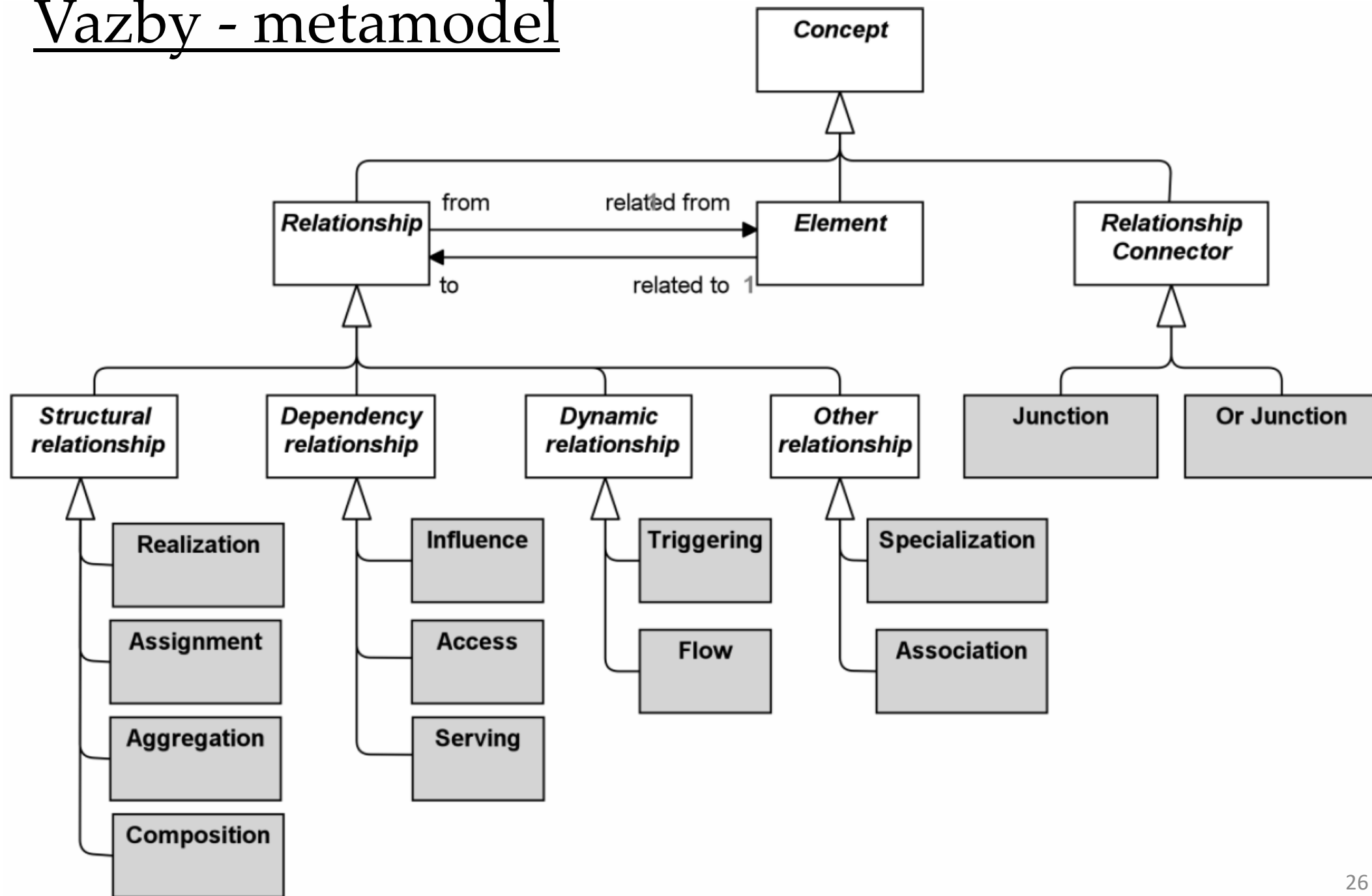
From → ↓ To		Application				
		Application Collaboration	Application Component	Data Object	Application Event	Application Function
Motivation	Assessment	n o	n o	n o	n o	n o
	Constraint	r n o	r n o	r n o	r n o	r n o
	Driver	n o	n o	n o	n o	n o
	Goal	r n o	r n o	r n o	r n o	r n o
	Meaning	n o	n o	n o	n o	n o
	Outcome	r n o	r n o	r n o	r n o	r n o
	Principle	r n o	r n o	r n o	r n o	r n o
	Requirement	r n o	r n o	r n o	r n o	r n o
	Stakeholder	n o	n o	o	o	n o
Strategy	Value	n o	n o	n o	n o	n o
	Capability	r o	r o	r o	o	r o
	Course Of Action	r o	r o	r o	o	r o
Business	Resource	r o	r o	r o	o	o
	Business Actor	v tfo	v tfo	o	t o	v tfo
	Business Collaboration	v tfo	v tfo	o	t o	v tfo
	Contract	a o	a o	r o	a o	a o
	Business Event	v tfo	v tfo	o	t o	v tfo
	Business Function	rv tfo	rv tfo	o	t o	rv tfo
	Business Interaction	rv tfo	rv tfo	o	t o	rv tfo
	Business Interface	rv tfo	rv tfo	o	t o	v tfo
	Business Object	a o	a o	r o	a o	a o
	Business Process	rv tfo	rv tfo	o	t o	rv tfo
	Product	fo	fo	o	fo	fo
	Representation	a o	a o	o	o	a o
	Business Role	v tfo	v tfo	o	t o	v tfo
	Business Service	rv tfo	rv tfo	o	t o	rv tfo
Application	Application Collaboration	scg rv tfo	scg rv tfo	o	tfo	v tfo
	Application Component	scg rv tfo	scg rv tfo	o	tfo	v tfo
	Data Object	a o	a o	scg o	a o	a o
	Application Event	irv tfo	irv tfo	o	scg tfo	v tfo
	Application Function	irv tfo	irv tfo	o	tfo	scg v tfo
	Application Interaction	irv tfo	irv tfo	o	tfo	cg v tfo
	Application Interface	cg rv tfo	cg rv tfo	o	tfo	v tfo
	Application Process	irv tfo	irv tfo	o	tfo	cg v tfo
	Application Service	irv tfo	irv tfo	o	tfo	rv tfo

U dále popsaných vazeb jsou uvedeny typické příklady použití.

Pro seznam všech možných použití vazeb viz dokument ArchiMate 3.0 Specification, přílohu B – Relationship Tables

(a)ccess (c)omposition (f)low
i(n)fluence ass(o)ciation (r)ealization
a(g)gregation ass(i)gnment
(s)pecialization (t)riggering ser(v)ing

Vazby - metamodel

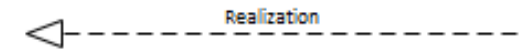


ArchiMate

Strukturální vazby (Structural Relationships)

Vazba – Realizace (Realization)

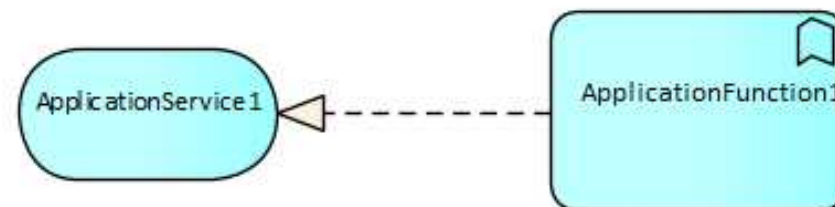
Typ: strukturální (Structural)



Definice:

Tato vazba říká, že entita ve směru šipky hraje kritickou roli při vytváření, existenci, dosažení cílů nebo činnosti , více abstraktní entity.

Příklad typického použití: aplikační služba (Application Service) je realizována aplikační funkcí (Application Function). Business služba (Business Service) je realizována business procesem (Business Process), nebo business funkcí (Business function) apod.



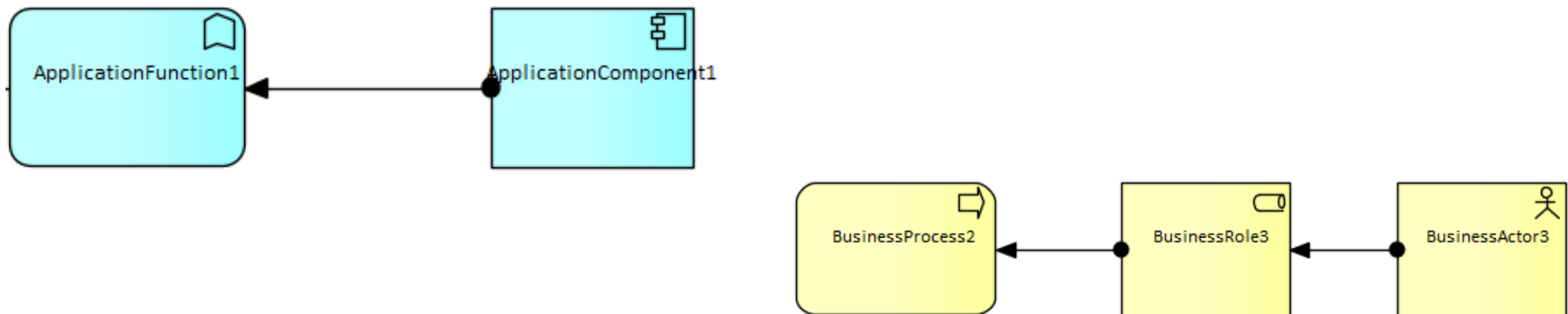
Vazba – Přiřazení (Assignment)

Typ: strukturální (Structural)

Definice:

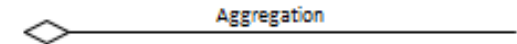
Tato vazba přiřazuje určité chování (Behavior) aktivnímu elementu (Active Structure). Vazba má od ArchiMate 3.0 směrovost a je tažena vždy od aktivního elementu k elementu, který reprezentuje chování. V určitých případech může být tato vazba natažena i mezi aktivními elementy, například od aktéra (Business Actor) k business roli (Business role).

Příklad typického použití: Aplikační funkce (Application Function), která reprezentuje chování, je přiřazena k aplikační komponentě, která je aktivním elementem.



Vazba – Agregace (Aggregation)

Typ: strukturální (Structural)

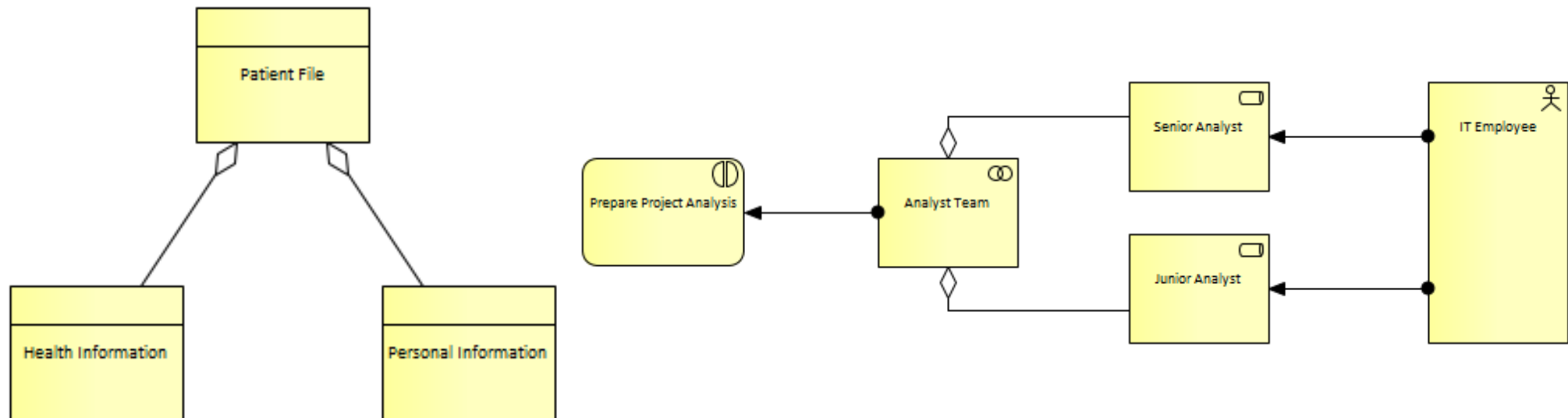


Definice:

Tato vazba definuje, že element k němuž je agregace vedena (na jehož straně se nachází diamond) se sestává z více jiných elementů. A tedy že element, od něž je agregace vedena, je součástí nějakého celku. Agregace je vždy povolena mezi elementy stejného typu. ArchiMate dále definuje, mezi jakými typy elementů je tato vazba dovolena (viz přehled povolených vazeb).

Na rozdíl od kompozice, může být jeden element součástí více celků (Agregací).

Příklad typického použití: Přiřazení rolí do kolaborací, skladba datových objektů apod.



Vazba – Kompozice (Composition)

Typ: strukturální (Structural)

Definice:

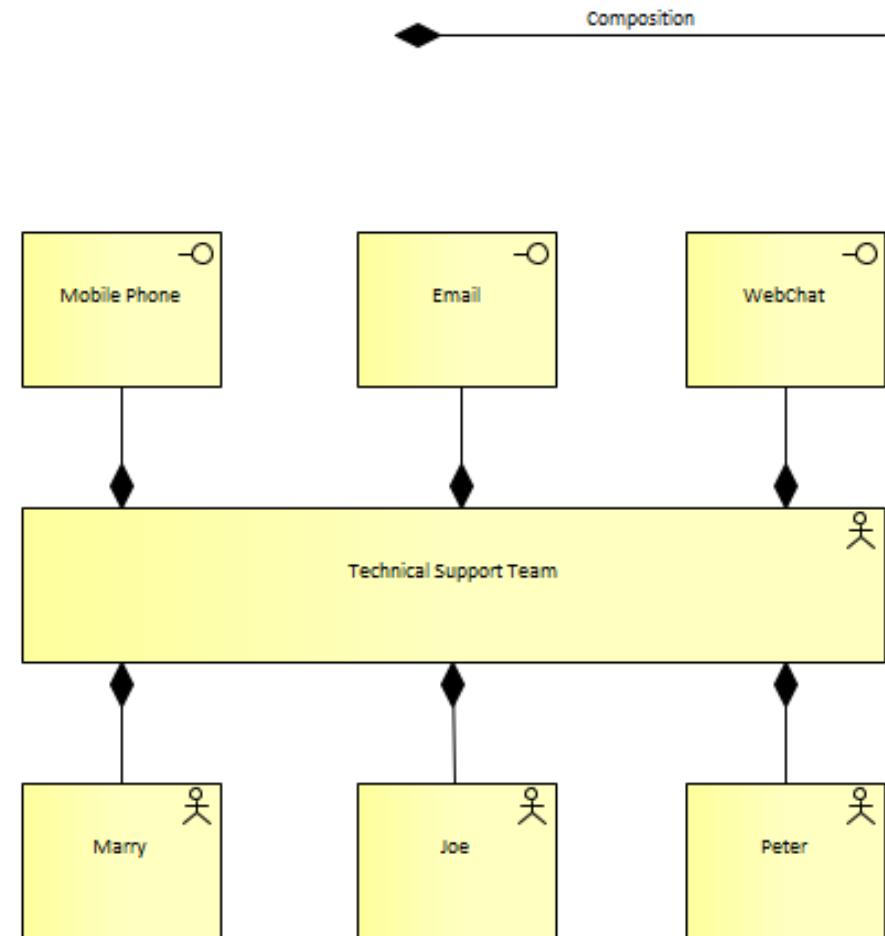
Tato vazba definuje vnitřní struktury (soudržnost) uvnitř architektury. Element, k němuž je kompozice vedena (na jehož straně se nachází diamond), se sestává z více jiných elementů. A tedy, že element od něž je agregace vedena, je součástí nějakého celku.

Kompozice (na straně diamondu) může být pouze element, sestávat se však může (na opačné straně vazby, než je diamond) i z vazeb (např. Communication Network). Poznámka – v EA 13 a výše lze propojovat vazby s vazbami, lze tedy tuto konstrukci použít.

Kompozice je vždy povolena mezi elementy stejného typu. ArchiMate dále definuje, mezi jakými typy elementů je tato vazba dovolena (viz přehled povolených vazeb).

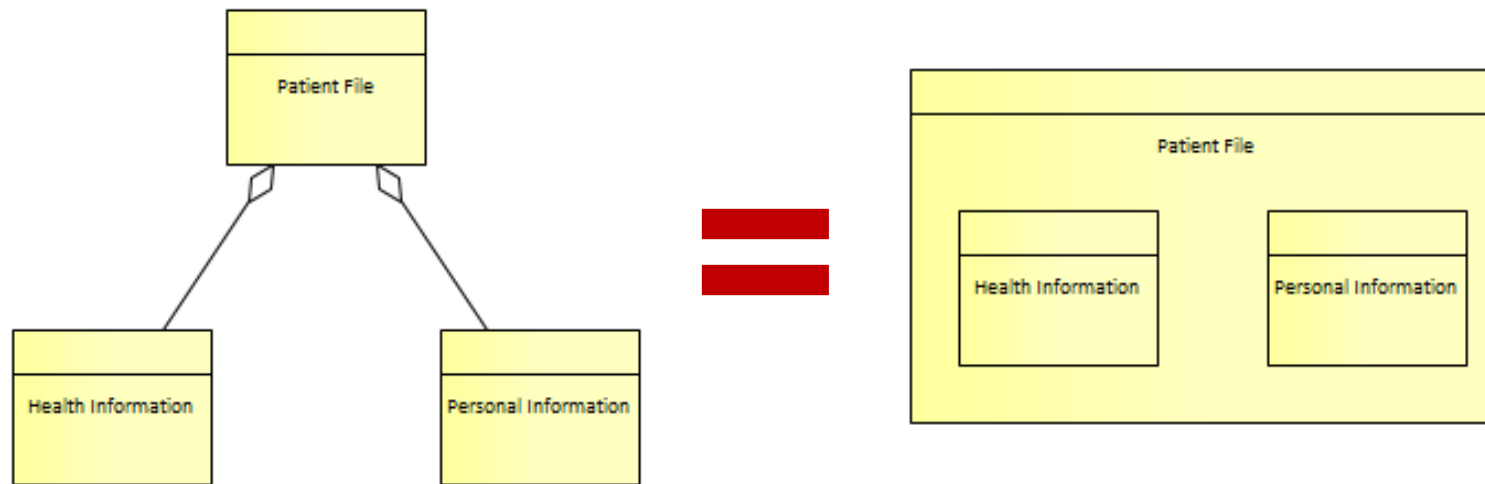
Jeden element nemůže být součástí více než jedné kompozice (to je rozdíl oproti UML).

Příklad typického použití: Rozpad komponenty aplikace do subkomponent. Přiřazení interface ostatním aktivním objektům. Přiřazení aktérů do týmů apod.



Nesting

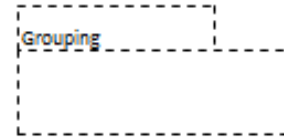
Existuje ještě další možnost jak vyjádřit, že některé elementy jsou navzájem ve strukturálním vztahu, tj. kompozice (Composition), agregace (Aggregation), přiřazení (Assignment) nebo realizace (Realization). Tento způsob se nazývá Nesting). Následující dva zápisy jsou, co do významu, shodné:



Poznámka: Nesting má ovšem jeden problém. Zápis pomocí Nestingu vypadá shodně pro assignment, realization, agregaci i kompozici a nelze na první pohled poznat, v jakém vztahu elementy jsou – lze to poznat pouze z kontextu a to ještě pouze tam, kde se nepoužívají všechny strukturální vazby (například role – collaboration).

Dále výslovně doporučuji (v Enterprise Architect) vytvářet příslušné vazby i v případě, že nebudou (kvůli Nestingu) na diagramech zobrazeny (pro dopadové analýzy, relationship matrix, traceability apod.).

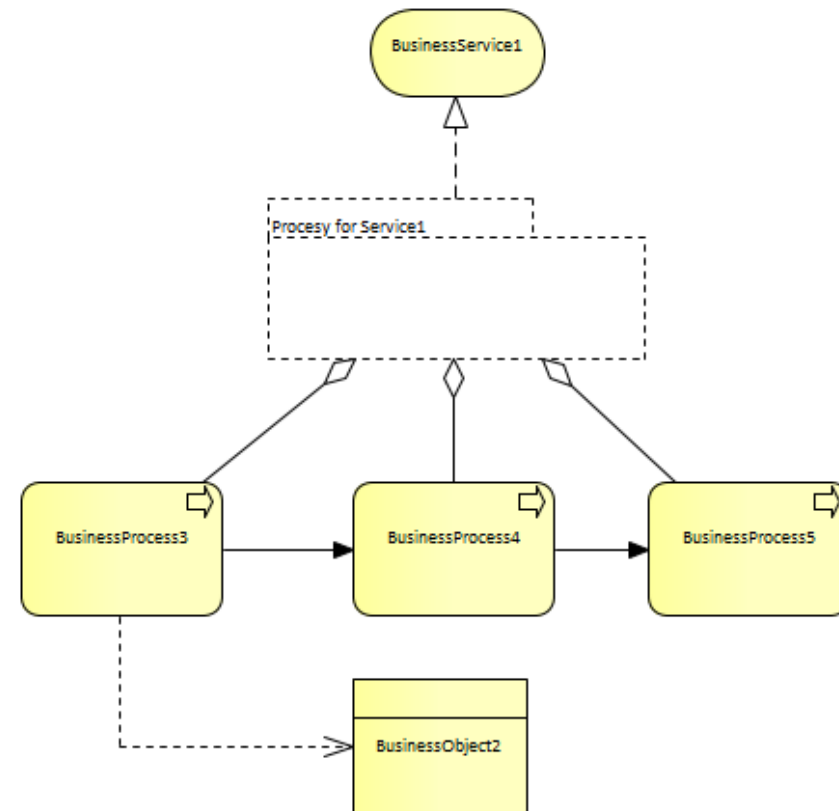
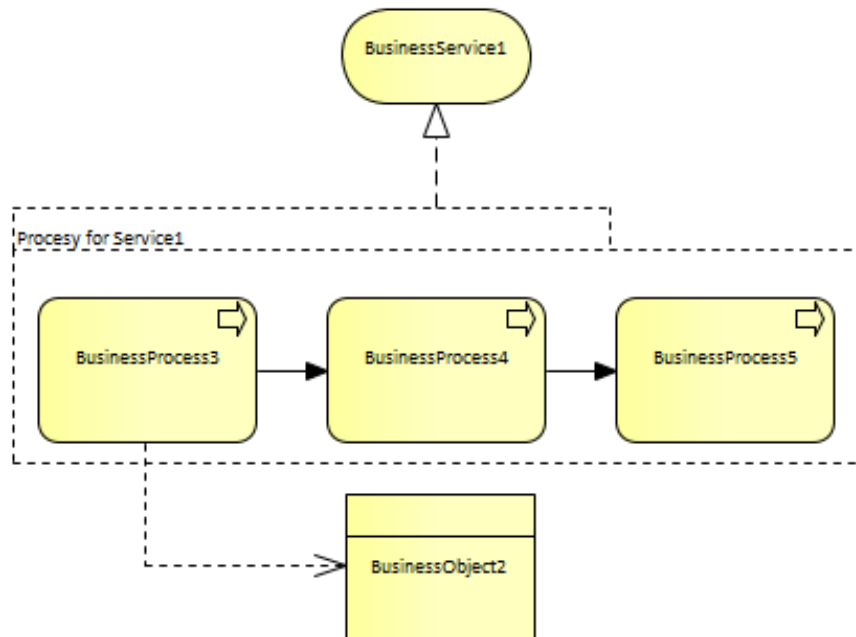
Skupinování (Grouping)



Grouping element agreguje (Aggregate nebo Composite) koncepty (elementy nebo vazby), které k sobě náležejí na základě nějaké sdílené charakteristiky.

Koncepty mohou náležet do více než jedné skupiny.

Existují dva způsoby zápisu skupiny – viz níže:



Poznámka: Grouping má ovšem jeden problém. Zápis pomocí Grouping vypadá shodně pro agregaci a kompozici a nelze na první pohled poznat, v jakém vztahu elementy jsou – lze to poznat pouze z kontextu a to ještě pouze tam, kde nelze použít obě vazby (například role – collaboration).

Dále výslovně doporučuji (v Enterprise Architect) vytvářet příslušné vazby i v případě, že nebudou (kvůli Grouping) na diagramech zobrazeny (pro dopadové analýzy, relationship matrix, traceability apod.).

ArchiMate

*Vazby definující závislost
(Dependency Relationships)*

Vazba – Vliv (Influence)

Typ: závislost (Dependency)

Definice:

Pomocí influence zaznamenáváme, že element (architektonický) ovlivňuje dosažení cíle motivačního elementu, například cíle (goal) nebo principu (principle). Tato vazba se rovněž používá pro znázornění, jak se motivační elementy ovlivňují mezi sebou.

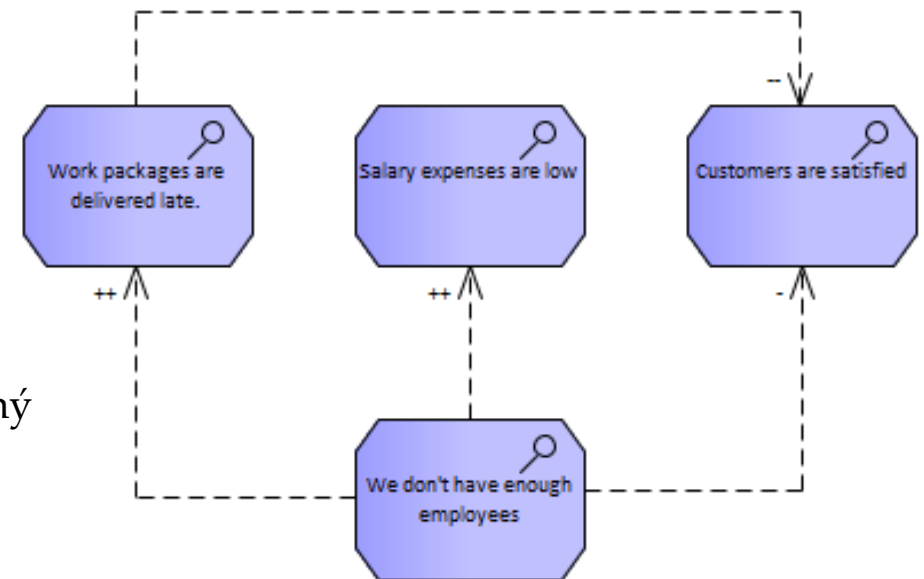
Vazba se používá od ovlivňujícího elementu k ovlivněnému a může být definována zda ovlivňuje element kladně nebo záporně (sign) a případně jakou silou. Defaultně je vazba bez těchto informací a znamená, že ovlivňující element „nějak“ ovlivňuje element po směru šipky:

Sign (zda ovlivňuje záporně nebo kladně): + -

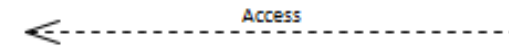
Síla: {++, +, 0/bez sign, -, --} – alternativně lze sílu zapsat jako [0..10], tedy třeba +3

Jeden element může ovlivňovat jiný element kladně a ještě jiný záporně.

Příklad typického použití: zaznamenání, jakým způsobem ovlivňuje komponenta (systém / aplikace) cíl společnosti (goal) zvýšení obrátu.



Vazba – Přístup (Access)



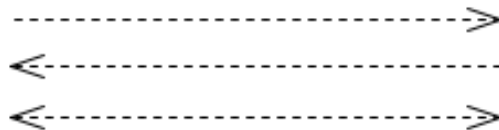
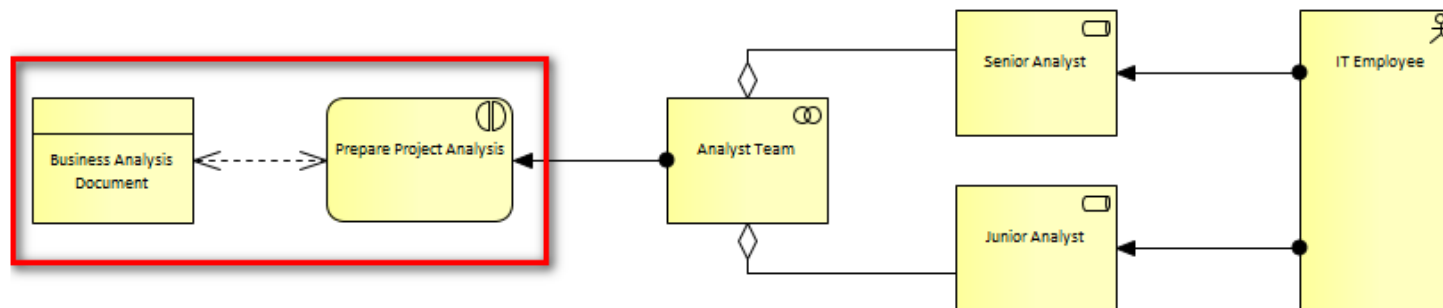
Typ: závislost (Dependency)

Definice:

Pomocí vazby Access modelujeme schopnost elementů typu chování (Behavior) a nebo aktivních elementů (Active structure) nahlížet nebo ovlivňovat pasivní elementy (Passive Structure).

Ovlivňováním se rozumí například vytváření, změna nebo mazání.

Vazba se táhne vždy směrem K pasivnímu objektu, nicméně šipka sama může znázorňovat, zda z objektu čteme, zapisujeme nebo obojí. V Enterprise Architect toto provádíme přímo na vazbě, volbou Direction



Access Properties

General
Role(s)
Constraints

Access Properties

General

Role(s)

Constraints

Direction: Source -> Destination

Style: Custom

Stereotype: ArchiMate_Access

Connector Properties

Source: BusinessRole1

Target: BusinessRole1

Alias:

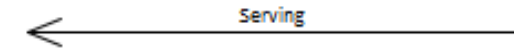
Direction: Source -> Destination

Style: Custom

Stereotype: ArchiMate_Access

Vazba – Serving

Typ: závislost (Dependency)

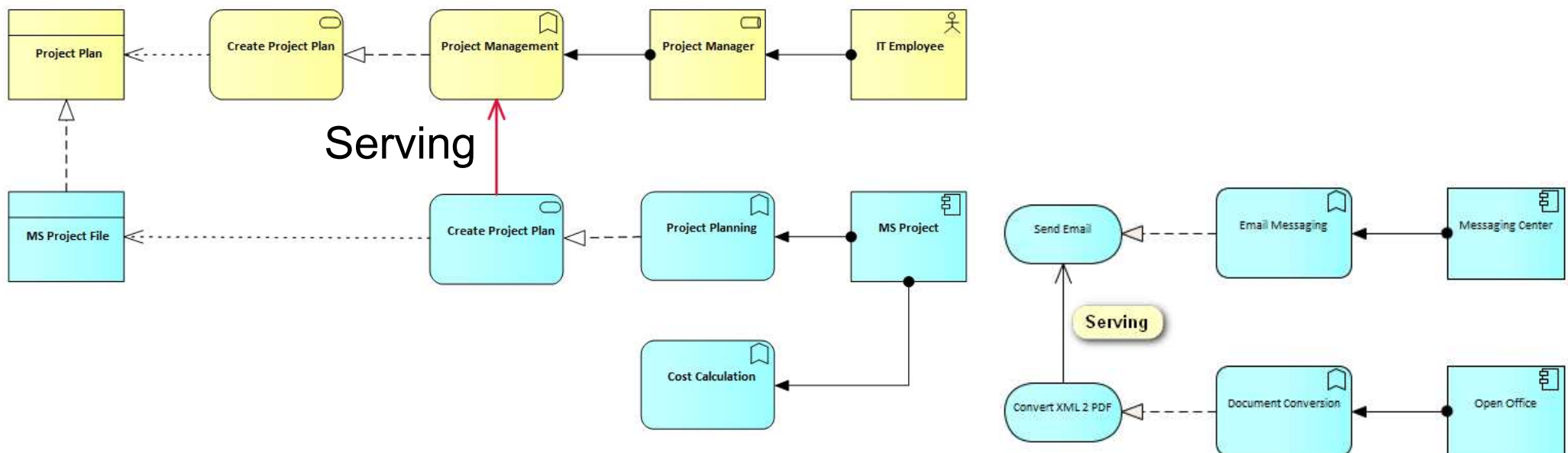


Definice:

Pomocí této vazby znázorníme, že jeden element poskytuje svou funkcionalitu jinému. Vazba je tažena vždy od elementu, který poskytuje svou funkcionalitu k elementu, jež danou funkcionalitu využívá.

Tato vazba se typicky používá pro přechod mezi vrstvami, lze ji ale použít i v rámci jedné vrstvy.

Příklad typického použití: aplikační služba nebo funkce (z Application layer) poskytuje svou funkcionalitu (nebo také podporuje) business procesu nebo business interakci (z Business layer). Dalším příkladem může být, že aplikační služba nebo funkce poskytuje svou funkcionalitu jiné aplikační službě nebo funkci.



ArchiMate

Dynamické vazby
(*Dynamic Relationships*)

Vazba – Tok (Flow)

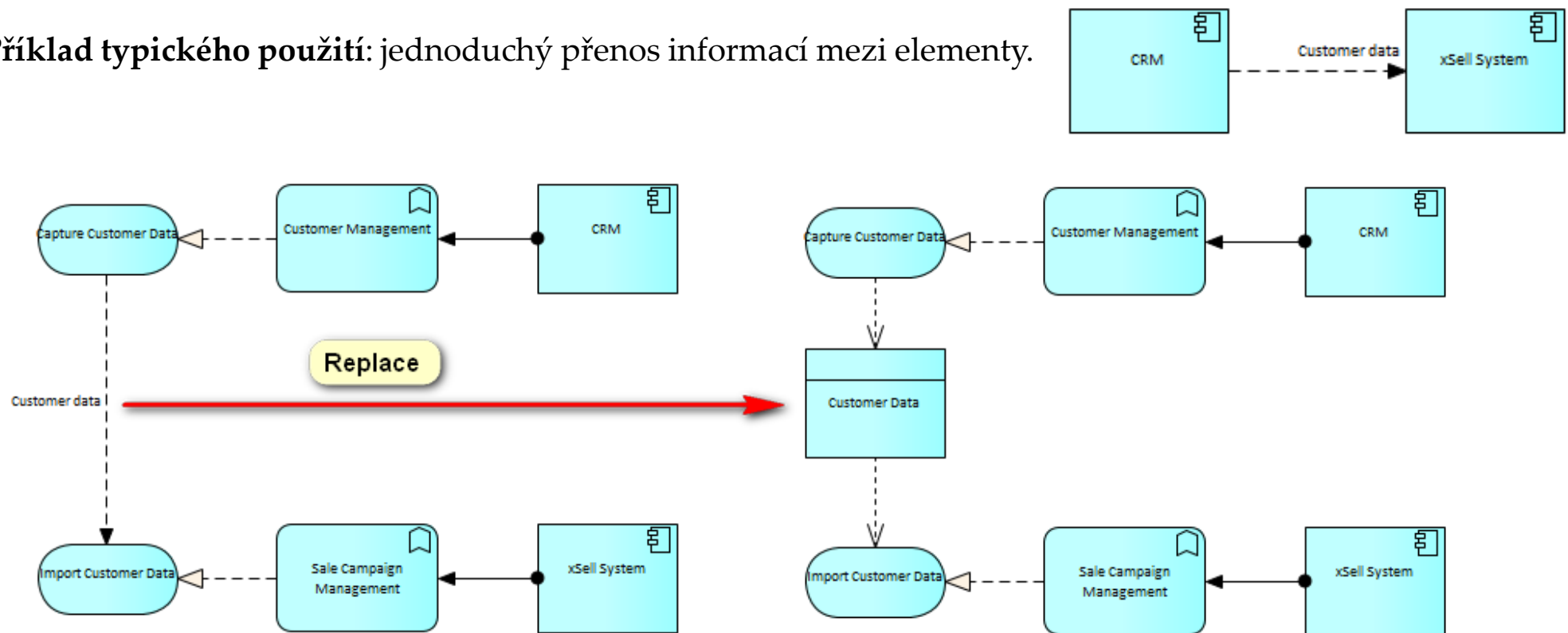
Typ: dynamické (Dynamic)

Definice:

Tato vazba reprezentuje přenos mezi dvěma elementy. Přenášeny mohou být informace, zboží či peníze. Vazba se používá typicky mezi elementy chování (Behaviour), nicméně lze ji použít i například pro znázornění přenosu informací mezi komponentami.

Touto vazbou se dá nahradit (za cenu ztráty detailu) například zápis, kdy dvě aplikační funkce přistupují k datovému objektu, kdy jedna ho vytvoří a druhá čte či konzumuje (consume) / používá.

Příklad typického použití: jednoduchý přenos informací mezi elementy.



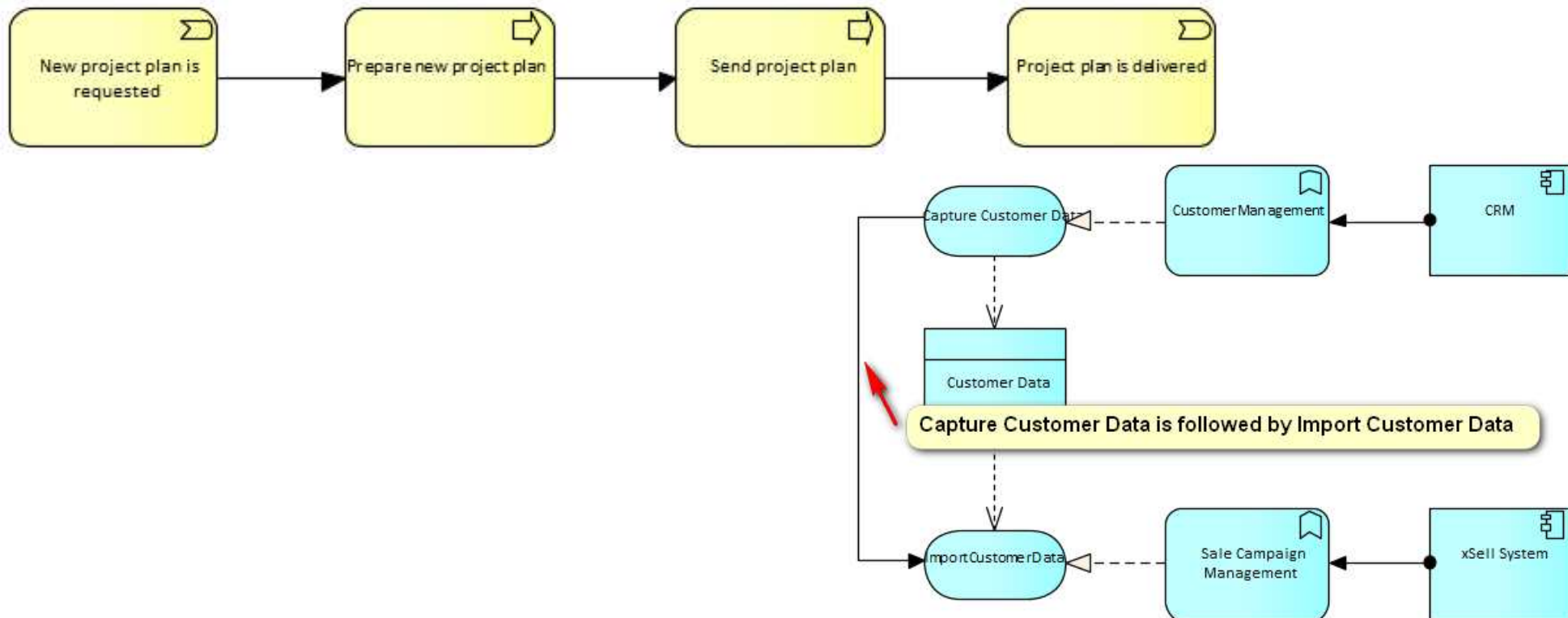
Vazba – Přechod (Triggering)

Typ: dynamické (Dynamic)

Definice:

Tato vazba popisuje pořadí činnosti mezi elementy. Je vykládána tak, že činnost elementu, z něhož vede vazba, by měla být ukončena dříve, než je zahájena činnost elementu (typicky chování – Behavior), k němuž vede vazba (po směru šipky).

Příklad typického použití: přechody mezi business procesy, business interakcemi (Business Interactions), událostmi (Events), ale třeba i aplikačními funkcemi.

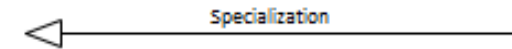


ArchiMate

*Ostatní vazby
(Other Relationships)*

Vazba – Specializace (Specialization)

Typ: jiné (Other)

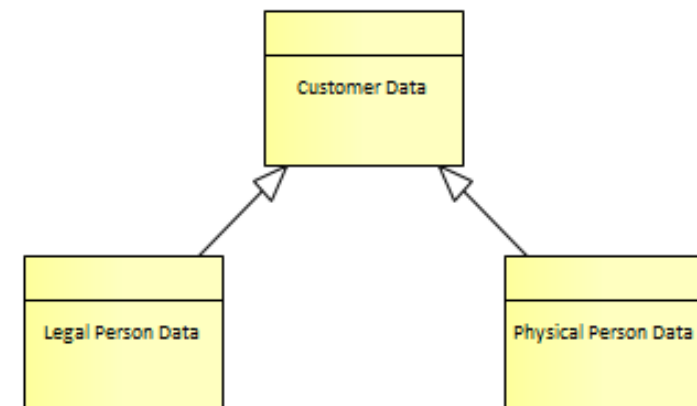
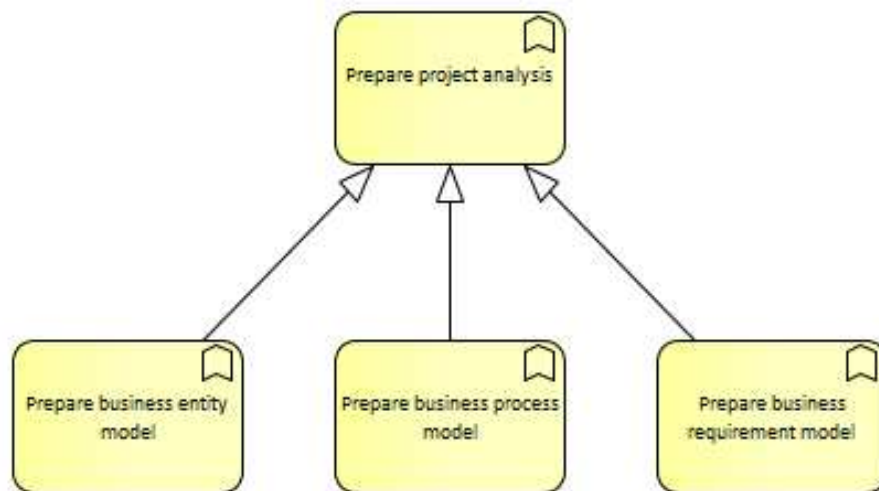


Definice:

Specializace říká, že specializovaný element (na konci vazby bez šipky) je speciální variantou (či druhem) elementu po směru šipky (na konci vazby se šipkou).

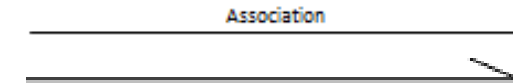
Tato vazba je inspirována vazbou generalizace / specializace v UML, umožňuje však její širší použití. Touto vazbou lze svázat jakékoliv dva koncepty (většinou elementy, ale i vazby) stejného typu.

Příklad typického použití: kdekoliv, kde je potřeba odvodit z obecného elementu více konkrétní.



Vazba – Asociace (Association)

Typ: jiné (Other)

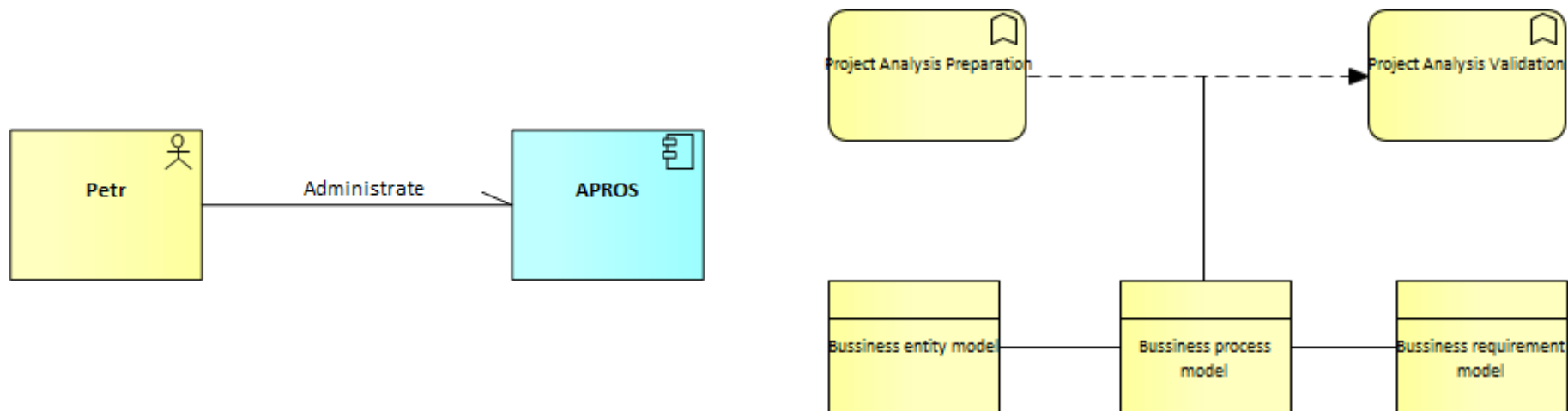


Definice:

Tuto vazbu použijeme, pokud chceme vyjádřit vztah mezi koncepty (elementy či vazbami) a nelze použít jinou vazbu. Association sama o sobě neříká, jaký vztah mezi elementy je - to je nutné specifikovat přímo na konkrétní vazbě.

Association by se měla používat co nejméně.

Příklad typického použití: kdekoliv, kde chceme vyjádřit vztah mezi elementy či vazbami a nemůžeme použít jinou vazbu.



Vazba – Spojení (Junction)

Typ: jiné (Other)

Definice:

Spojení (Junction) se používá pro propojení dvou vazeb stejného typu. Nejedná se o vazbu jako takovou, spíše o konektor k jiné vazbě (vazbám).

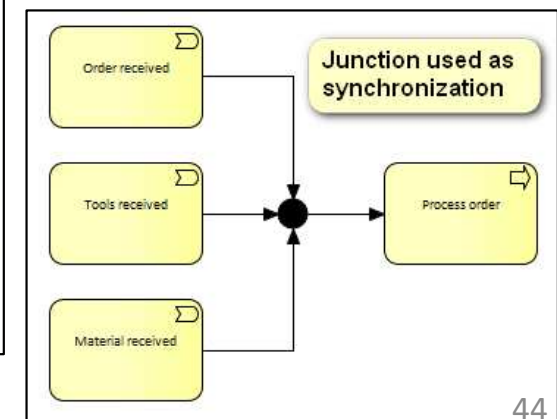
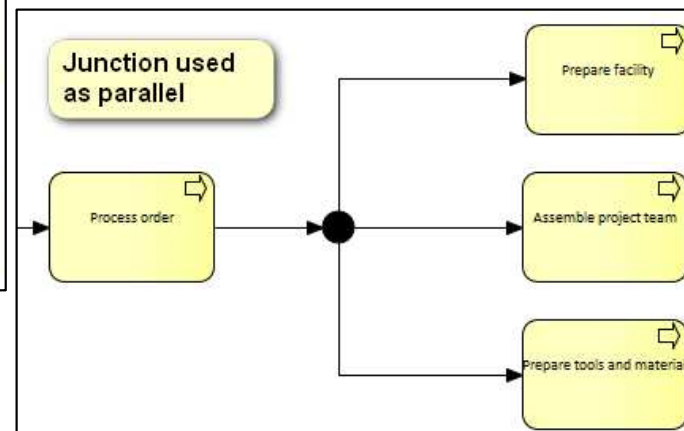
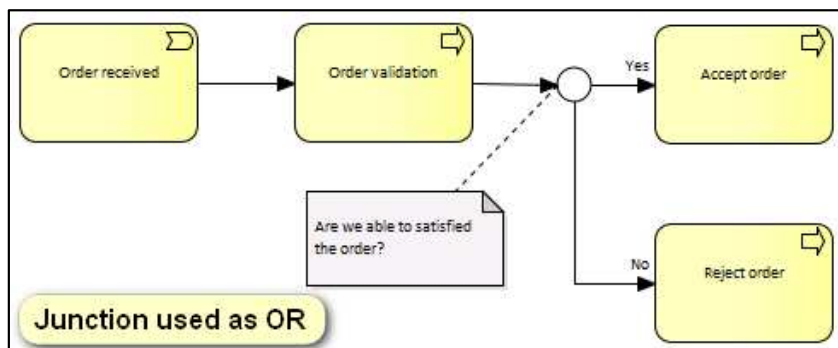
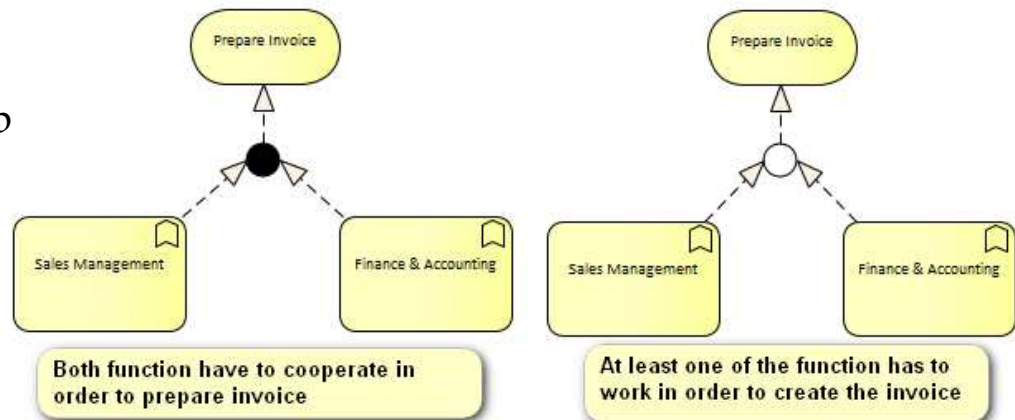
Může být typu AND (a) nebo OR (nebo).

Příklad typického použití: podmínky, paralelizace a synchronizace v business procesech. Dalším příkladem je definice, že dvě aplikační funkce musí spolupracovat, aby naplnily požadovaný účel.

Junction - AND



Junction - OR



ArchiMate

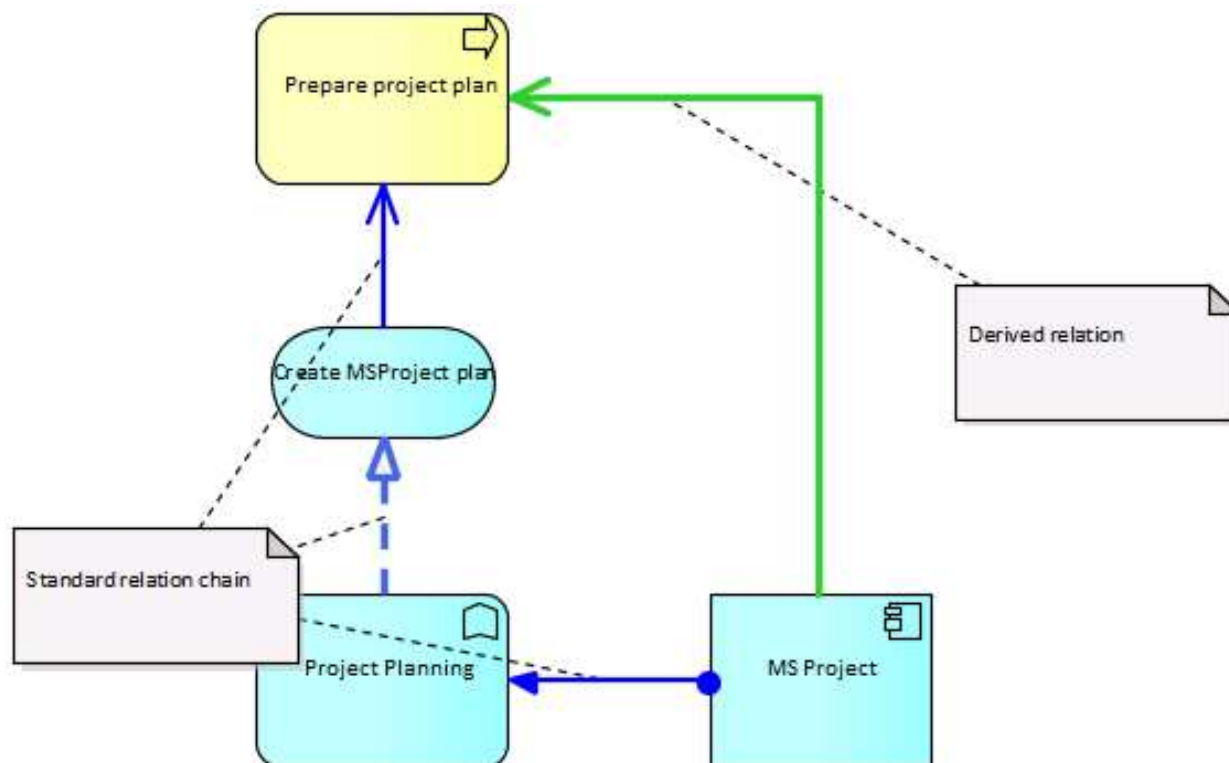
Derivační pravidla vazeb
(*Derivation Rules*)

Derivace vazeb – základní definice

Derivací vazeb se rozumí odvození přímé vazby mezi elementy z řetězce vazeb nepřímých.

Příklad: pokud máme business proces, který je podporován (pomocí serving) aplikační službou, ta je realizována aplikační funkcí (realized) a ta je přiřazena (jako chování – Behavior) aplikační komponentě, pak můžeme propojit onu aplikační komponentu přímo s procesem, pomocí vazby serving.

Derivovaná vazba může být jen tak silná, jako nejslabší vazba v řetězci!



Derivace vazeb – proč derivovat vazby

Proč bychom mohli chtít derivovat vazby:

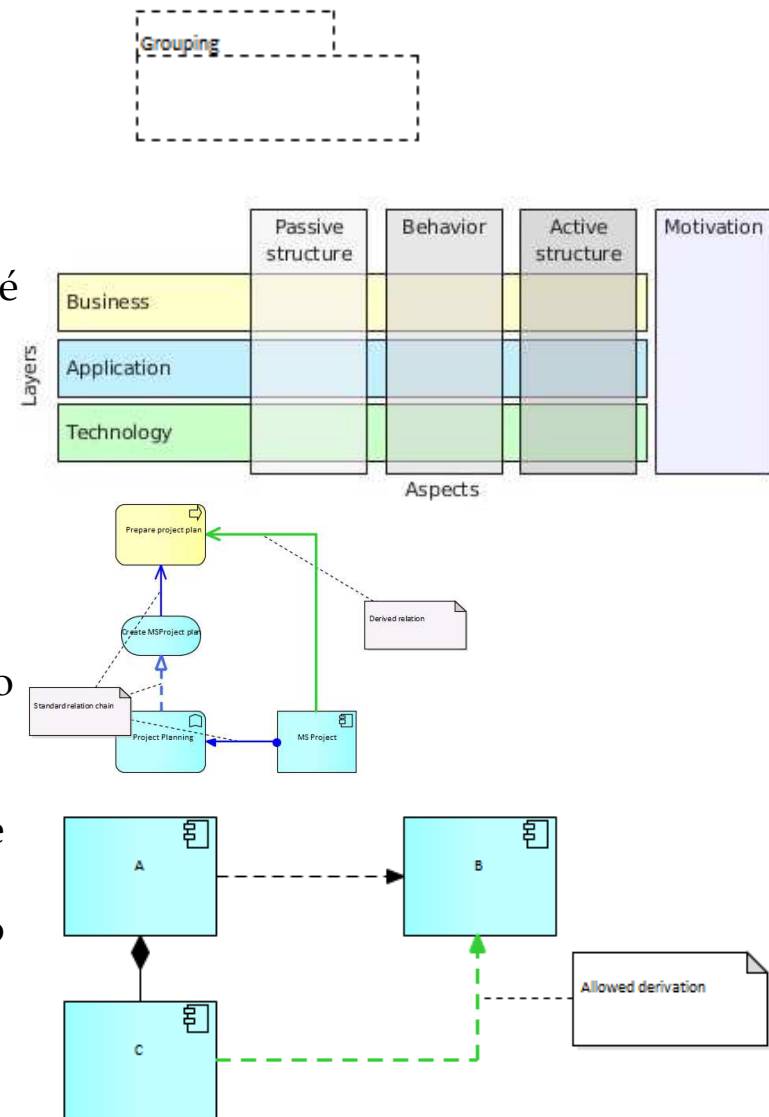
- Abstrahování od detailů – v některých případech potřebujeme zobrazit závislost elementů na sobě, aniž bychom nutili čtenáře procházet celou hierarchii vazeb. Například můžeme chtít znázornit závislost business procesu (Business layer) na hardware (Technology layer), aniž bychom chtěli zobrazit celý stromček závislostí (business proces – application service – application function – technology service – technology function – device – node).
- Některé pohledy na architekturu (Viewpoints – bude probráno dále) vyžadují použití derivovaných vazeb.
- Nedostatek času – někdy není, například v rámci projektu, čas na provedení kompletní analýzy.
- Dopadová analýza – některé analytické nástroje (a Enterprise Architect je mezi nimi) mají užitečné funkcionality, které umožňují rychlou dopadovou analýzu, založenou na propojení elementů vazbami. Přímé propojení elementů pomocí derivovaných vazeb využití těchto funkcí značně zjednodušuje.

Target +	ArchiMate 3 View::Application	ArchiMate 3 View::Application	ArchiMate 3 View::Application	ArchiMate 3 View::Application	ArchiMate 3 View::BusinessAct	ArchiMate 3 View::BusinessOb	ArchiMate 3 View::BusinessPrc	ArchiMate 3 View::BusinessRol	ArchiMate 3 View::DataObject	ArchiMate 3 View::Node1	ArchiMate::Derivace vazeb	ArchiMate::Layers	ArchiMate::Příklad 2	ArchiMate::Příklad projektu	ArchiMate::TOGAF	ArchiMate::Základní prvky	Business Entities::IntersectingE
+ Source																	
ArchiMate 3 View::Busines...																	
ArchiMate 3 View::Busines...																	
ArchiMate 3 View::Busines...																	
ArchiMate 3 View::Busines...																	
ArchiMate 3 View::DataObj...																	
ArchiMate 3 View::Node1																	
ArchiMate::Derivace vazeb																	

Derivace vazeb – pravidla

Derivace vazeb podléhá určitým pravidlům:

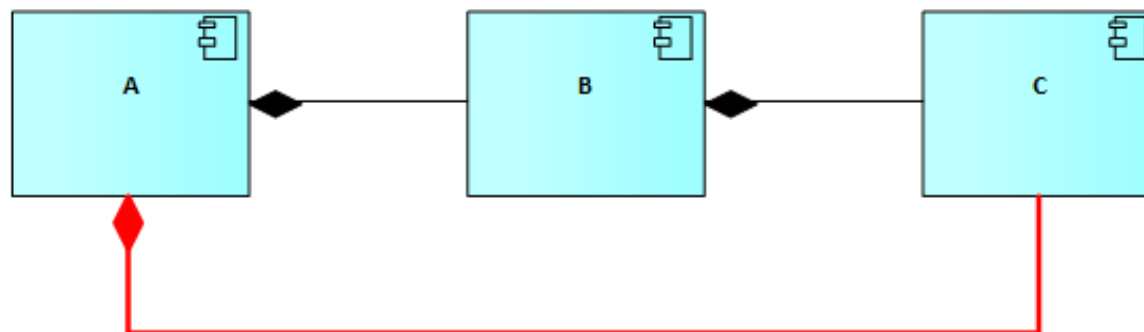
- Nelze derivovat vazby, které využívají (jsou navázané) na element grouping.
- Nelze derivovat vazby mezi core elementy (patřící do jedné ze tří základních vrstev – Business, Application a technology) a mezi elementy patřící do jedné z extenzí (Motivation, Strategy a Implementation). Výjimkou z tohoto pravidla je, pokud elementy z core a extenzí jsou provázané vazbou realizace nebo influence.
- Pokud derivujeme řetězec vazeb mezi elementy, pak použijeme pro derivaci vždy nejslabší vazbu z řetězce nebo slabší.
- Pokud vede Flow z elementu A do elementu B a zároveň je element C součástí elementu A (pomocí agregace nebo kompozice), pak je povoleno rovněž Flow z elementu C do elementu B. Obdobně, pokud by byl element C navázan pomocí agregace nebo kompozice na element B, bylo by povolené Flow z A do C.



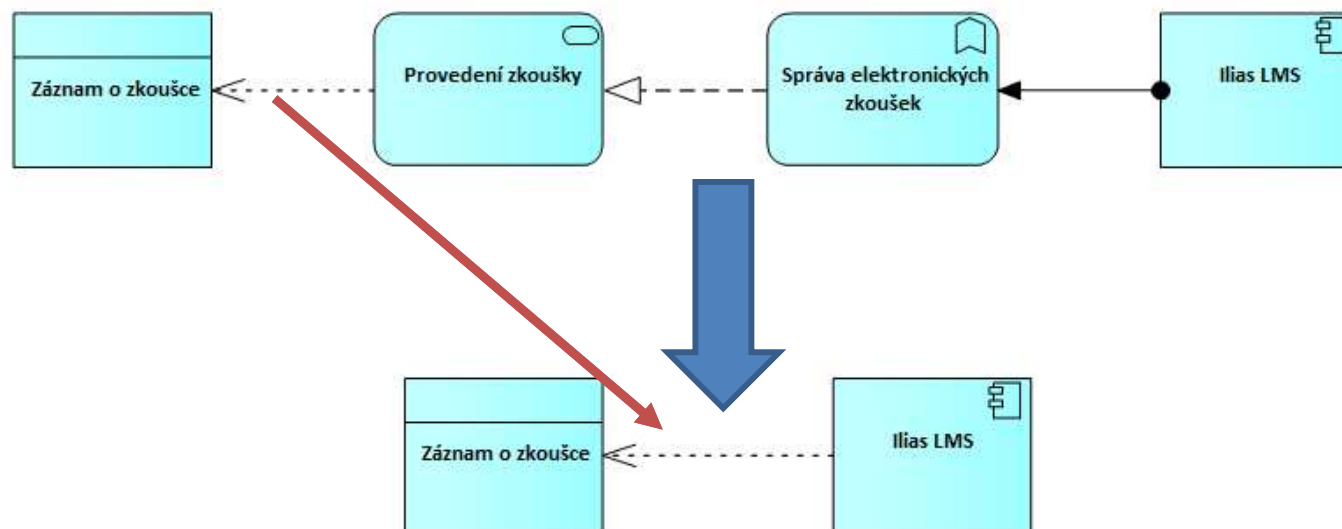
Pokud si nejste jisti, mezi kterými elementy je povolena jaká vazba, ověřte si platnost zamýšlené vazby v dokumentu ArchiMate 3.0 Specification, příloha B – Relationship Tables. Pro další informace o derivovaných vazbách viz stejný dokument, kapitola 5.6. Ne všechny, zde navržené derivace, je však vhodné používat, např. Example 17: Derived Triggering Relationship je vyložené nevhodný.

Derivace vazeb – příklad

Příklad implicitní derivace – mezi komponentou A a C je implicitní kompozitní vazba



Příklad derivace mezi komponentou a daty



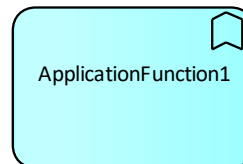
ArchiMate

Elementy

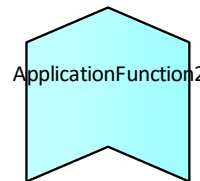
Způsob zápisu elementů

Elementy je možné zapsat dvěma způsoby:

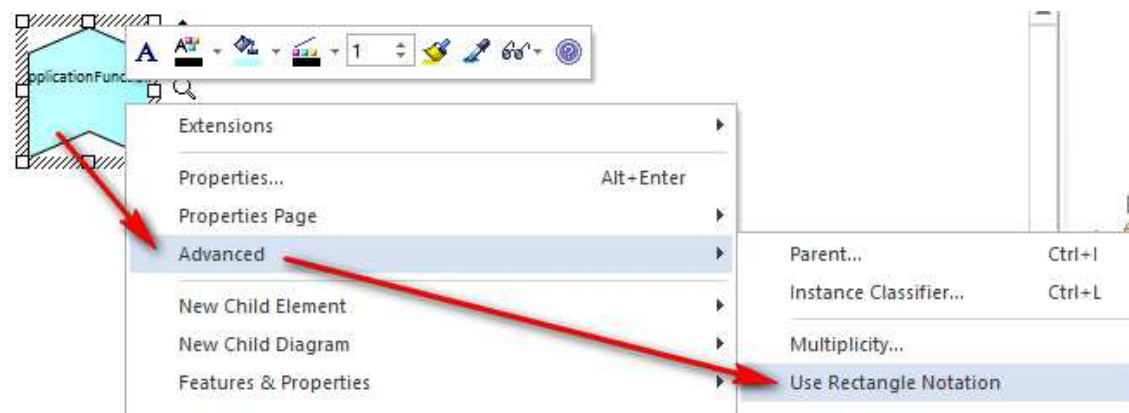
- Klasický obdélníkový zápis s ikonkou v pravém horním rohu.



- Přímě ve tvaru ikonky.



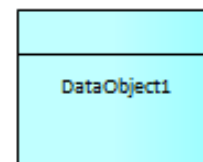
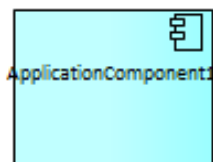
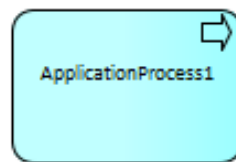
Oba způsoby zápisu jsou z hlediska ArchiMate rovnocenné. Výhodou ikonkového zápisu může být, že čtenář spíše nepřehlédne význam elementu, nevýhodou (kromě poněkud dětské podoby diagramů) že analytik neustále musí u všech elementů (v EA) překlíkávat zobrazení nastavení – nelze jej jednoduše nastavit jako default (bez customizace). Velmi často se nechávají ve tvaru ikonky Business Actor a Services (Business, Application a Technology)



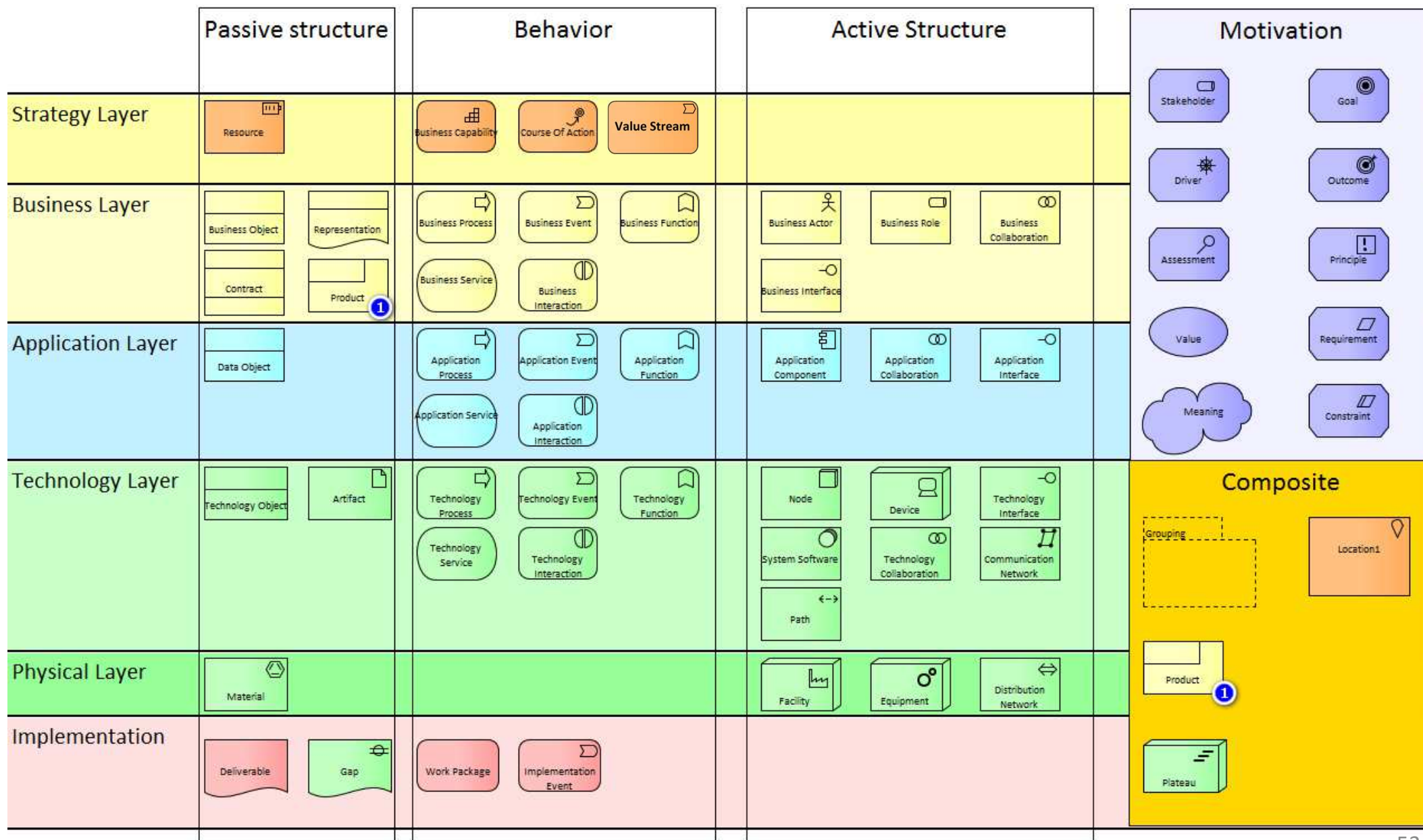
Rozlišení elementů podle rohů

Dobrou pomůckou pro identifikaci typů objektů jsou jejich rohy

- Ostré rohy – aktivní elementy 
- Zaoblené rohy – elementy chování (behavior) 
- Ostrý roh s přeškrtnutím – pasivní elementy (pozor, ne všechny to tak mají – například artifact vypadá jinak) 
- Zkosené rohy – motivační element 



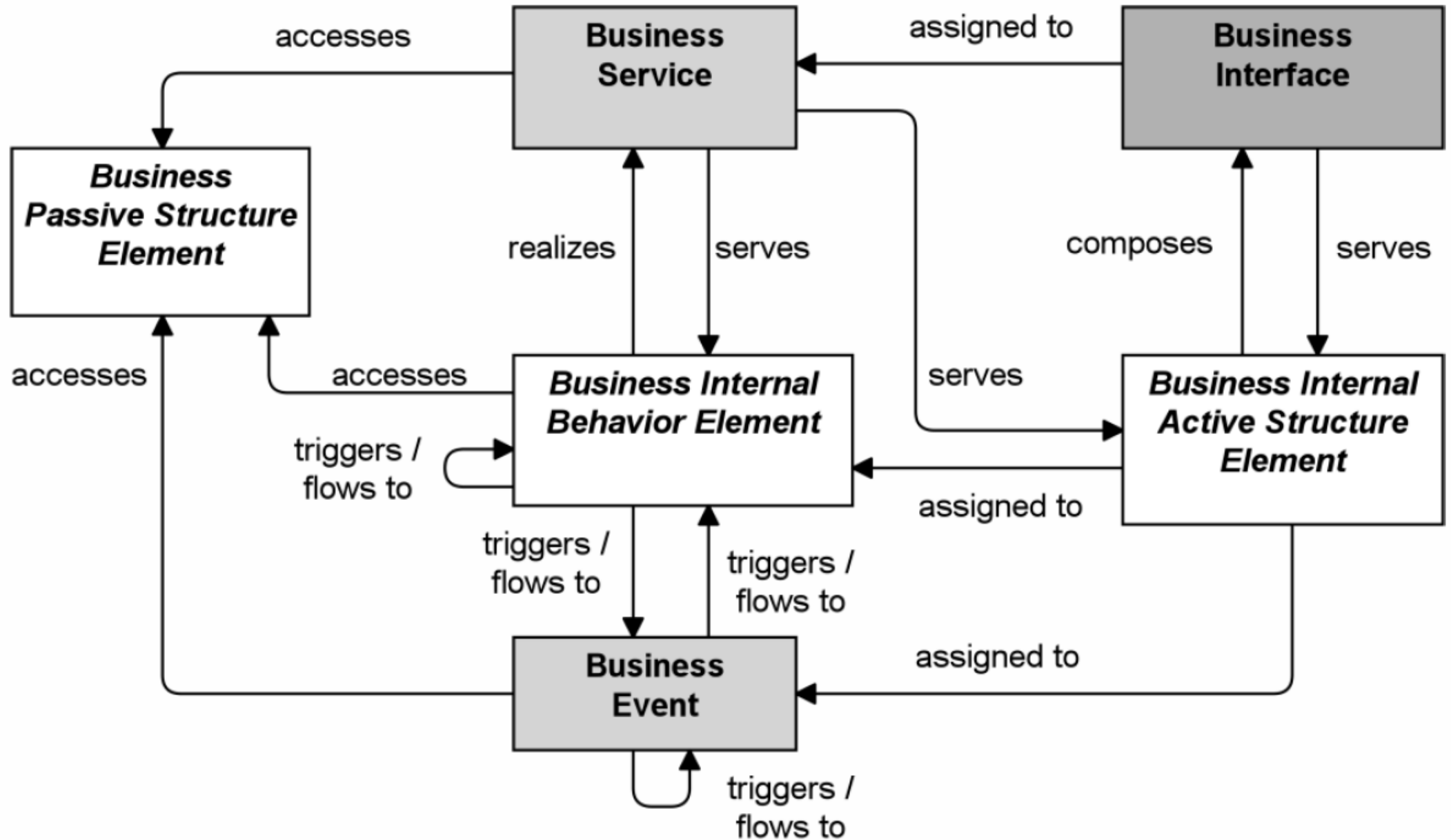
Velký přehled elementů po vrstvách a aspektech



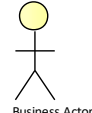
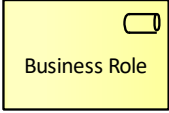
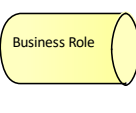
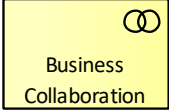
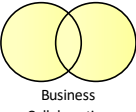
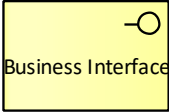
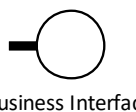
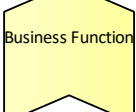
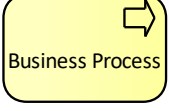
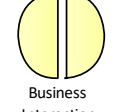
ArchiMate

Elementy – Business Layer

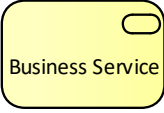
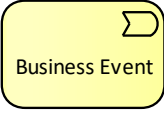
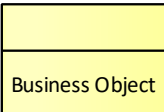
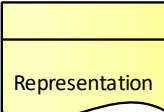
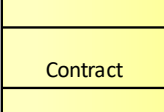
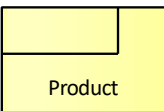
Business Layer - metamodel



Elementy v Business Layer – přehled #1

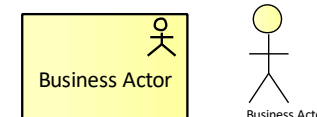
Element	Definice	Aspekt
 	A business actor is a business entity that is capable of performing behavior.	Active structure
 	A business role is the responsibility for performing specific behavior, to which an actor can be assigned, or the part an actor plays in a particular action or event.	Active structure
 	A business collaboration is an aggregate of two or more business internal active structure elements that work together to perform collective behavior.	Active structure
 	A business interface is a point of access where a business service is made available to the environment.	Active structure
 	A business function is a collection of business behavior based on a chosen set of criteria (typically required business resources and/or competences), closely aligned to an organization, but not necessarily explicitly governed by the organization.	Internal Behavior
 	A business process represents a sequence of business behaviors that achieves a specific outcome such as a defined set of products or business services.	Internal Behavior
 	A business interaction is a unit of collective business behavior performed by (a collaboration of) two or more business roles.	Internal Behavior

Elementy v Business Layer – přehled #2

Element	Definice	Aspekt
 	A business service represents an explicitly defined exposed business behavior.	External Behavior
 	A business event is a business behavior element that denotes an organizational state change. It may originate from and be resolved inside or outside the organization.	Behavior
	A business object represents a concept used within a particular business domain.	Passive Structure
	A representation represents a perceptible form of the information carried by a business object.	Passive Structure
	A contract represents a formal or informal specification of an agreement between a provider and a consumer that specifies the rights and obligations associated with a product and establishes functional and non-functional parameters for interaction.	Passive Structure
	A product represents a coherent collection of services and/or passive structure elements, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers. <i>Poznámka: tento element je často uváděn jako součást Business Layer, nicméně podle ArchiMate superstruktury patří do kompozitních elementů, podobně jako location nebo grouping. Proto je uveden v obou přehledech.</i>	Passive Structure Composite

Elementy v Business Layer – Actor #1

Typ: aktivní (Structural Active)



Definice:

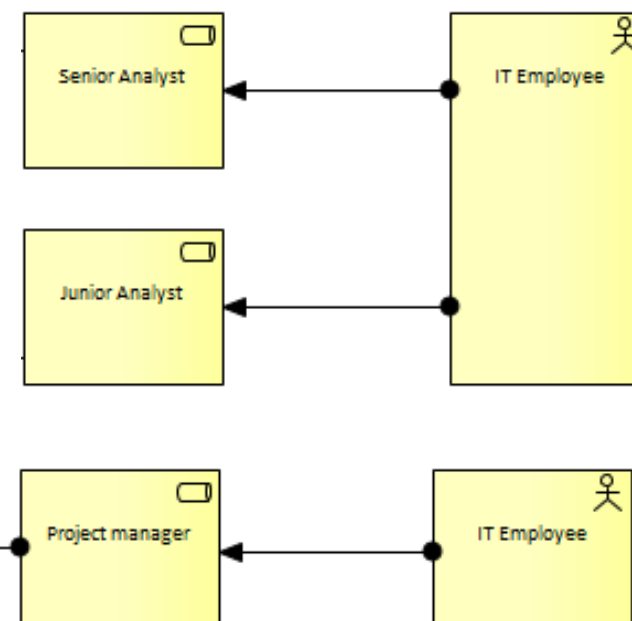
Business aktér (Business Actor) je entita, která je schopná určitého chování.

Definice :

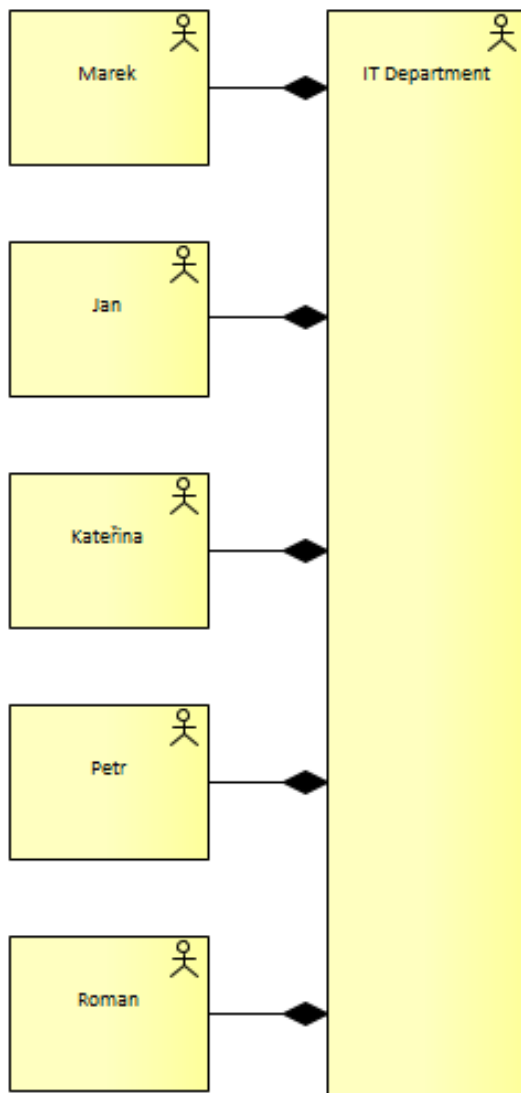
- Aktéři mohou být interní (zaměstnanci) nebo externí – zákazníci, dodavatelé, partneři.
- Aktéři mohou reprezentovat konkrétního člověka, oddělení nebo organizaci a to na různé úrovni abstrakce. Nemůže však (na rozdíl třeba od UML) reprezentovat technologickou komponentu nebo systém.
- Aktéři mohou být přiřazeny k rolím, které dále vymezují odpovědnost za určité chování.
- Aktéři se velmi často komponují pomocí agregací a kompozicí.

Příklady business actor:

- Pracovník oddělení RISK
- Oddělení RISK
- Jiří Vomáčka
- Dodavatel (generický dodavatel)
- Rádi Dodáme s.r.o. (konkrétní dodavatel)

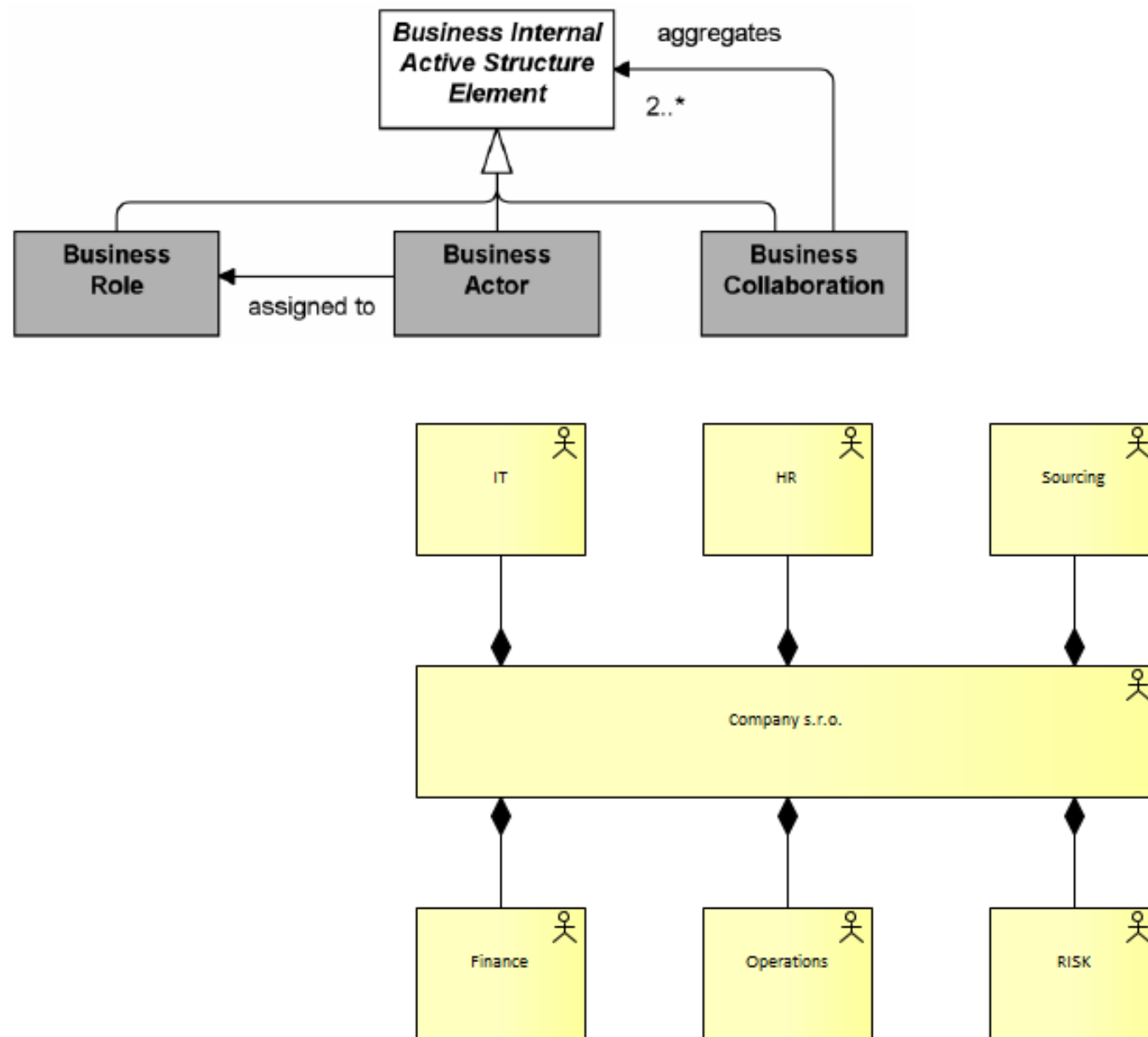


Elementy v Business Layer – Actor #2



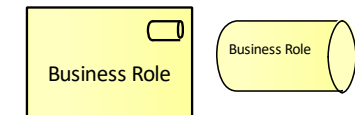
Příklad :

IT Oddělení je složeno z několika konkrétních osob.



Elementy v Business Layer – Role #1

Typ: aktivní struktura (Structural Active)



Definice:

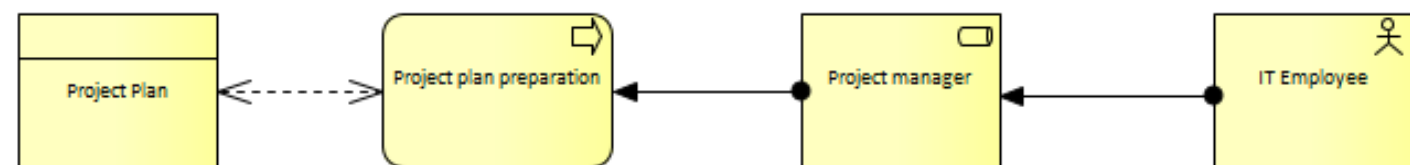
Business Role představuje odpovědnost za specifické chování, ke kterému může být aktér (Business Actor) přiřazen nebo za úlohu, kterou aktér hraje v akci či události.

Definice :

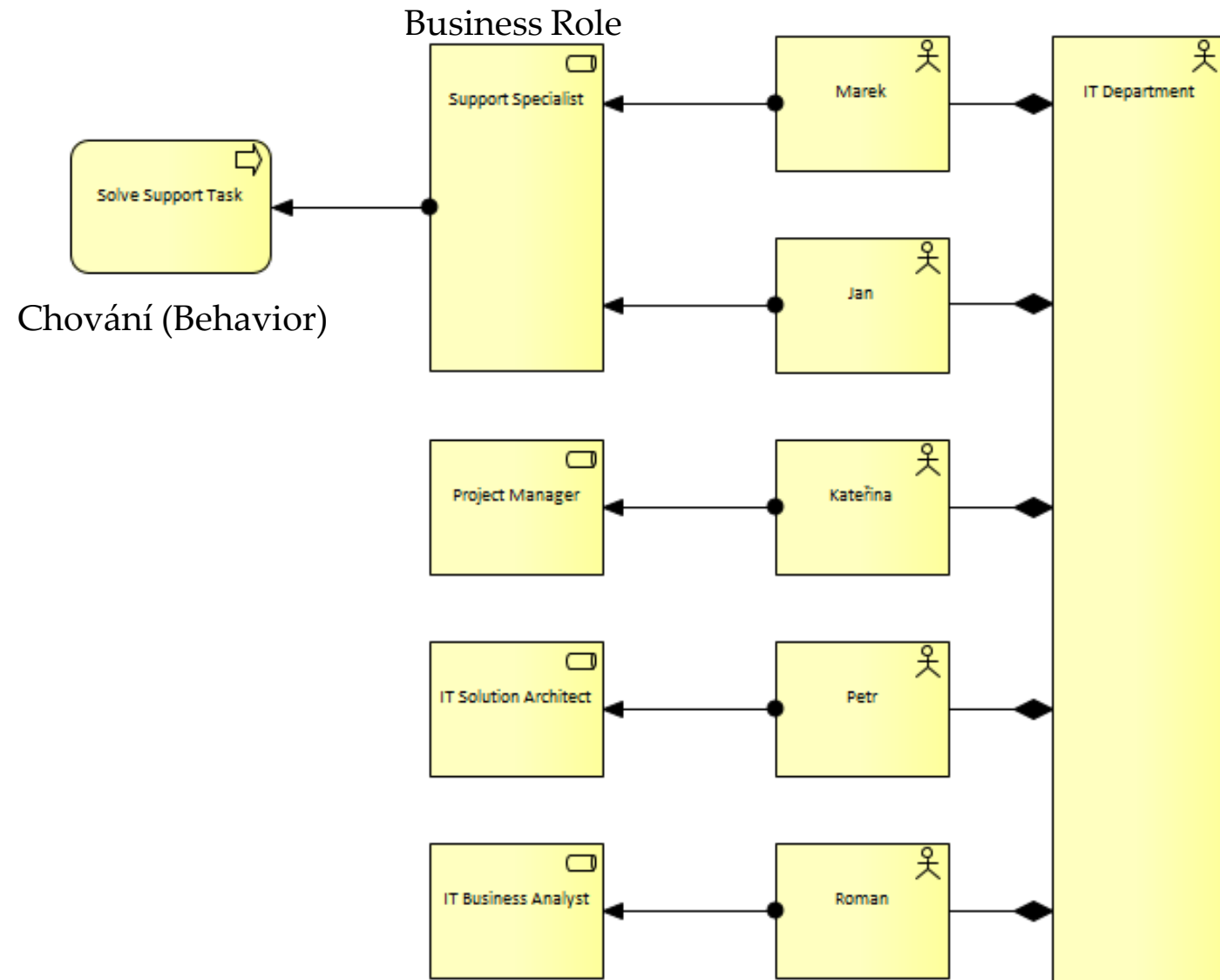
- Role představuje odpovědnost nebo schopnosti, které jsou přiřazeny určitému business procesu (Business Process) nebo business funkci (Business Function). Aktér (Business Actor), který je přiřazen této roli, je odpovědný za to, že odpovídající chování (business proces či business funkce) je provedeno a to buď tak, že ho aktér přímo provede, deleguje jej či jeho provedení řídí (manage).
- Roli lze dále využít při definování organizační struktury (obdobně jako aktér).
- Role může být přiřazena jedné či více business procesům / funkcím a jeden business aktér může být přiřazen i více než jedné roli.

Příklady business rolí:

- Project Manager
- Senior Specialista
- IT Analytik

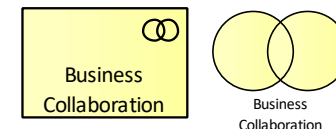


Elementy v Business Layer – Role #2



Elementy v Business Layer – Collaboration #1

Typ: aktivní struktura (Structural Active)



Definice:

Business kolaborace (Business Collaboration) agreguje dvě nebo více business aktivní struktury (Active Structure – aktéry nebo role), které musí spolupracovat aby dosáhly společného účelu (společného chování – behavior).

Definice :

Business proces nebo funkce mohou být často interpretovány jako chování jedné business role (či aktéra). V některých případech se však jedná o společné úsilí více než jedné role. Tato spolupráce (collaboration) může být ve výsledku více, než prostý součet schopností jednotlivých rolí. Business collaboration reprezentuje právě toto společné chování.

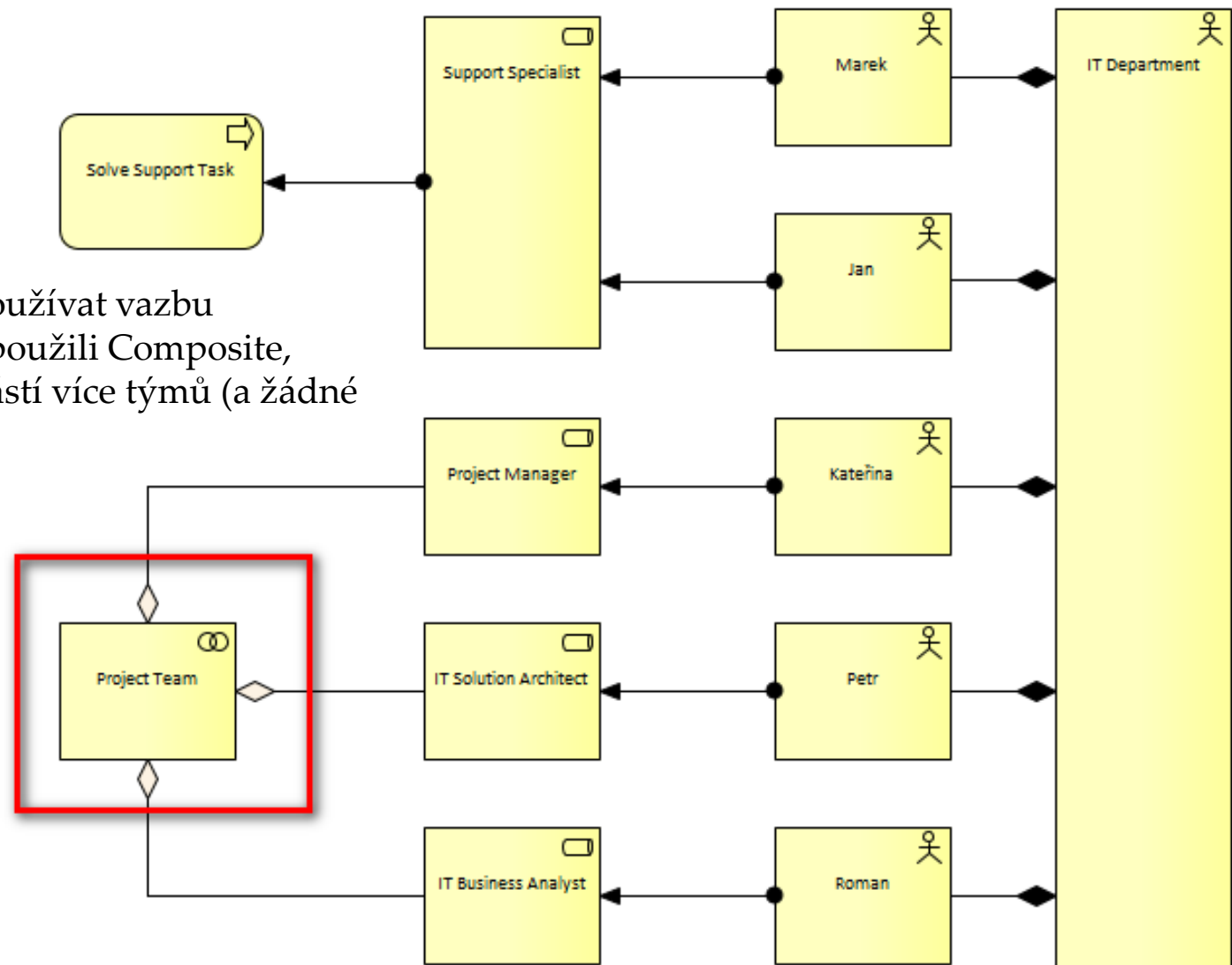
- Kolaborace mohou být dočasné – na rozdíl od vytváření organizační struktury pomocí aktérů a rolí, kolaborace nevyžadují trvalou strukturu uvnitř organizace a jsou poplatné jednomu či více konkrétních úkolů / účelů (viz slide Business Interaction).
- Kolaborace se často pojmenovávají podle týmu nebo účelu.

Příklady business kolaborací:

- Projektový tým
- Komise pro schválení rozpočtu stavby

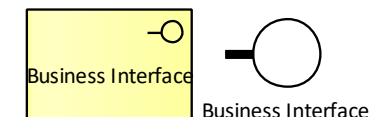
Elementy v Business Layer – Collaboration #2

Prakticky vždy bychom měli používat vazbu Aggregation – pokud bychom použili Composite, daná role by nemohla být součástí více týmů (a žádné jiné strukturální skladby)



Elementy v Business Layer – Interface #1

Typ: aktivní struktura (Structural Active)



Definice:

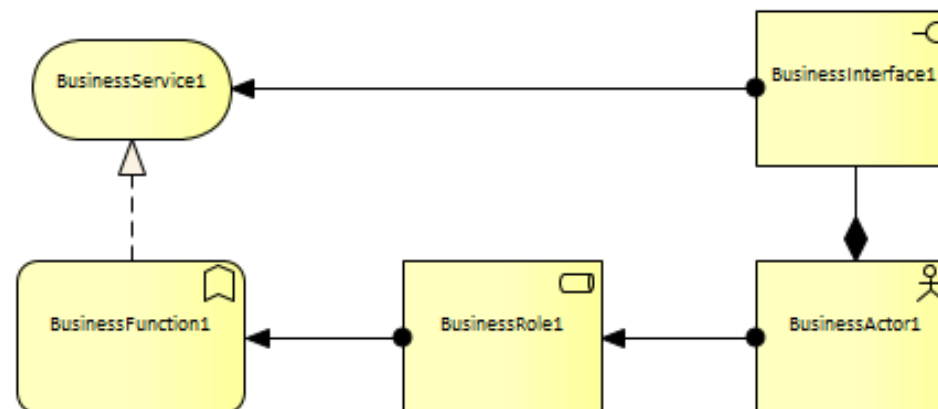
Business Interface je přístupový bod, kde je business služba dostupná z okolního prostředí.

Definice :

- Business Interface vystavuje funkcionalitu business služby jiným rolím nebo aktorům
- Business Interface bývá často označován jako kanál (telefon, internet, email, chat).
- Business Interface může být přiřazen jedné či více business službám (Business Service) pomocí vazby Assignment.
- Business Interface může být součástí business role nebo business aktéra pomocí vazby Composition nebo Aggregation.
- Business Interface může sloužit business roli (pomocí vazby Serving od role k interface).
- Název Business Interface by měl být podstatné jméno.

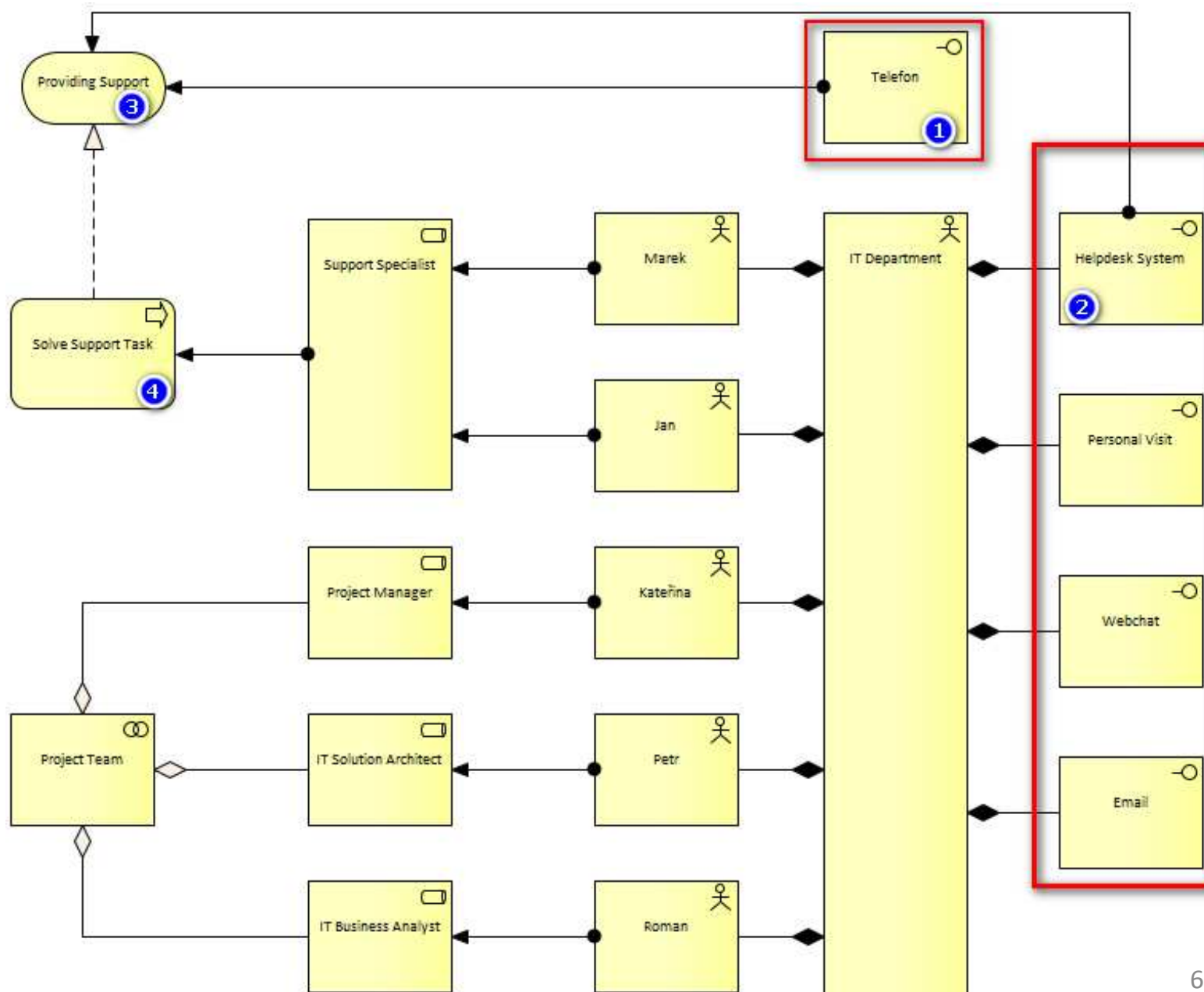
Příklady business interface:

- Telefon
- Email
- Webchat
- Fyzická přítomnost na pobočce

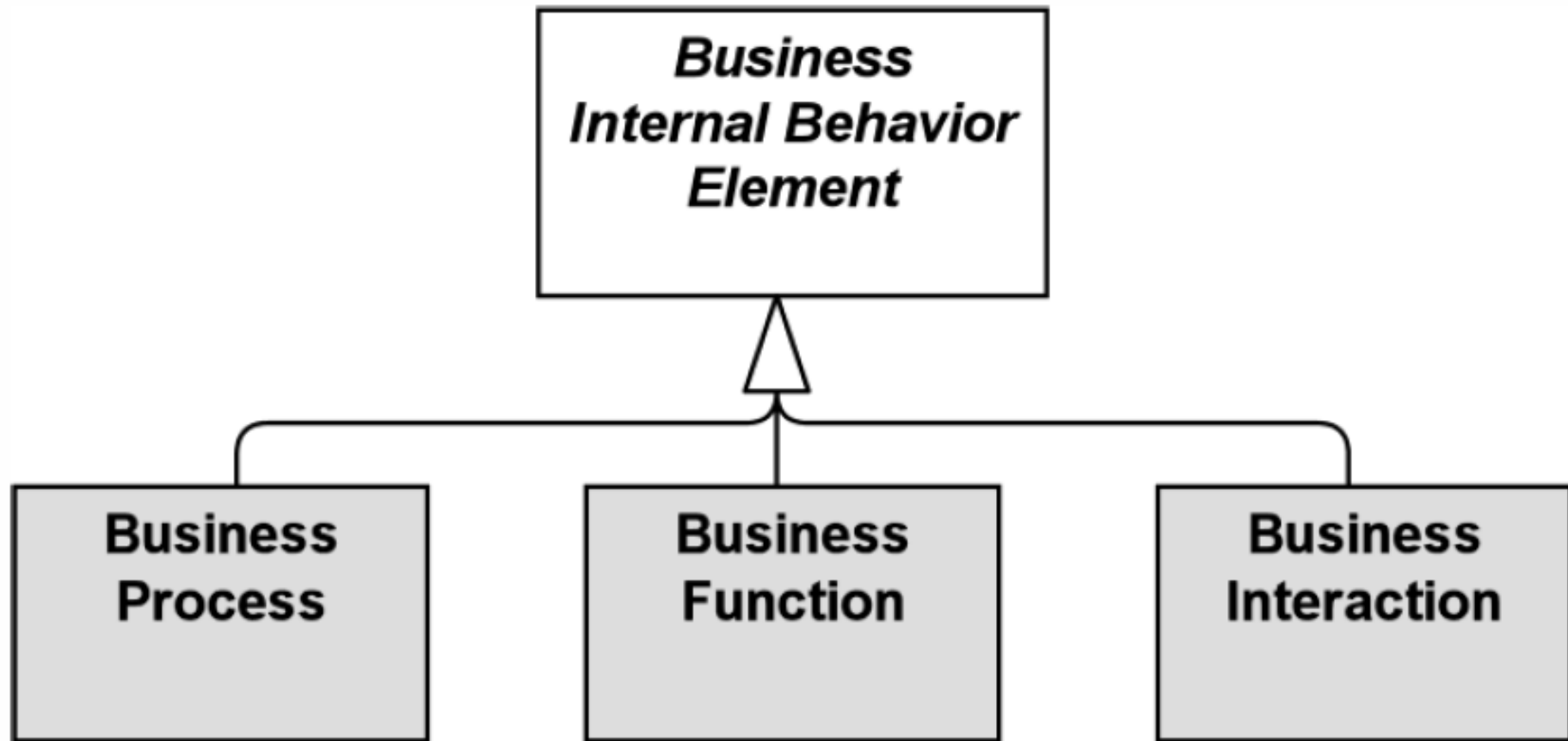


Elementy v Business Layer – Interface #2

1. Interface telefon není v rámci IT oddělení obecně dostupný kanál – je dostupný pouze v rámci business služby Poskytování podpory.
2. Helpdesk systém se sice používá pro službu Poskytování podpory (je to zdůrazněno), je to však obecně dostupný business interface.
3. Služba Poskytování podpory.
4. Je realizována pomocí procesu Vyřešení úkolu podpory.

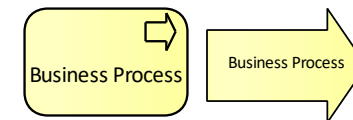


Elementy v Business Layer – Internal Behavior



Elementy v Business Layer – Business Process #1

Typ: aktivní interní (Internal Behavior)



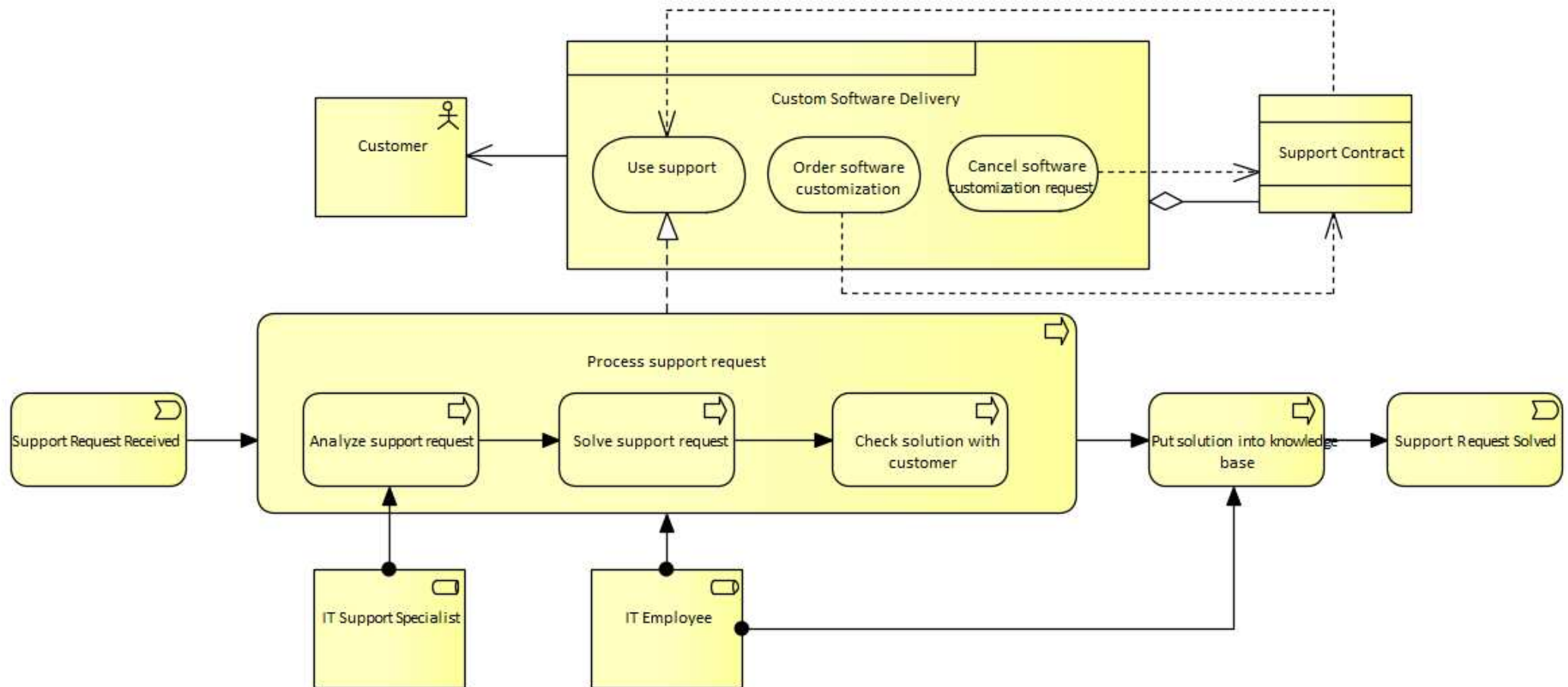
Definice:

Business proces reprezentuje sekvenci akcí (či chování – Behavior), které musí být splněny pro dosažení požadovaného výstupu, který je definován například jedním či více produkty (Business Product) nebo business služeb (Business Service).

Definice :

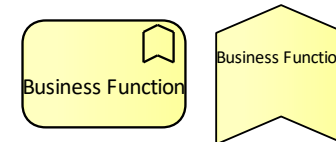
- Business proces může realizovat jednu, či více business služeb a náležet do jednoho, či více produktů. Z hlediska uživatele finální služby se jedná spíše o blackbox (proto internal).
- K business procesu může být přiřazena business role, která příslušný proces vykonává.
- Proces může agregovat jiné procesy na vyšší úrovni detailu. Těmto detailnějším procesům mohou být přiřazeny rovněž detailnější role.
- Mezi business procesy a business funkcemi může existovat mnoho M:N vztahů (asociací, agregací).
- Rozdíl mezi business procesem a business službou spočívá v tom, že proces je sekvence aktivit, kdežto business funkce (Business Function) seskupují činnosti podle požadovaných schopností, znalostí, zdrojů apod. Jinak řečeno, business funkce spíše odpovídá popisu schopnosti businessu provádět určitou činnost (business capability).
- *Pozn. je jen velmi málo situací, kdy nelze použít business proces a musí se použít business funkce (Business Function).*
- Business proces může být spouštěn business událostí (Business Event) nebo jiným business procesem a spouštět jinou událost nebo proces.
- Business proces může být podporován (Served – vazbou Serving ve směru šipkou k procesu) jiným business procesem, business funkcí, business službou. Business interakcí a obdobnými elementy z aplikační a technologické vrstvy.
- Business proces může přistupovat k business objektům (a dalším Business Passive strukturám).
- Název business procesu by měl vyjadřovat činnost (měl by obsahovat sloveso).

Elementy v Business Layer – Business Process #2



Elementy v Business Layer – Business Function #1

Typ: aktivní interní (Internal Behavior)



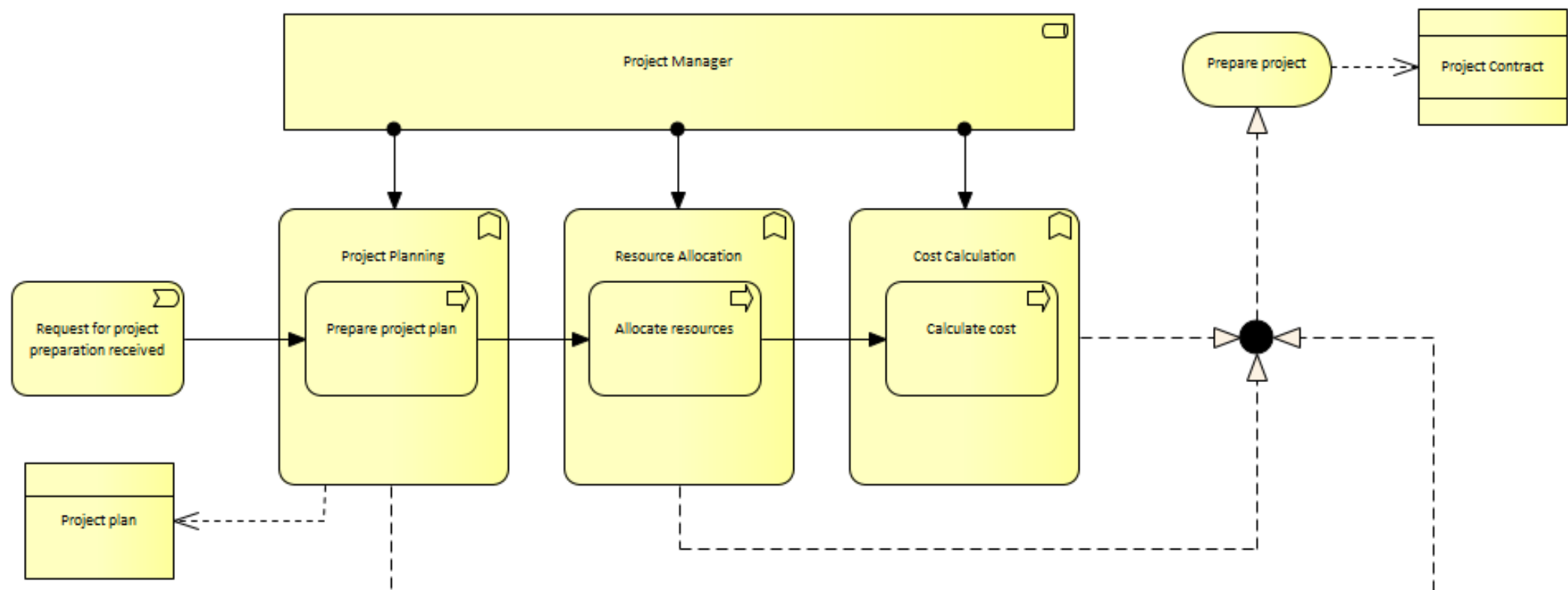
Definice:

Business funkce (Business Function) představuje soubor chování, který je definován na základě zvoleného množiny kritérií (například zdroje, business oblast nebo kompetence).

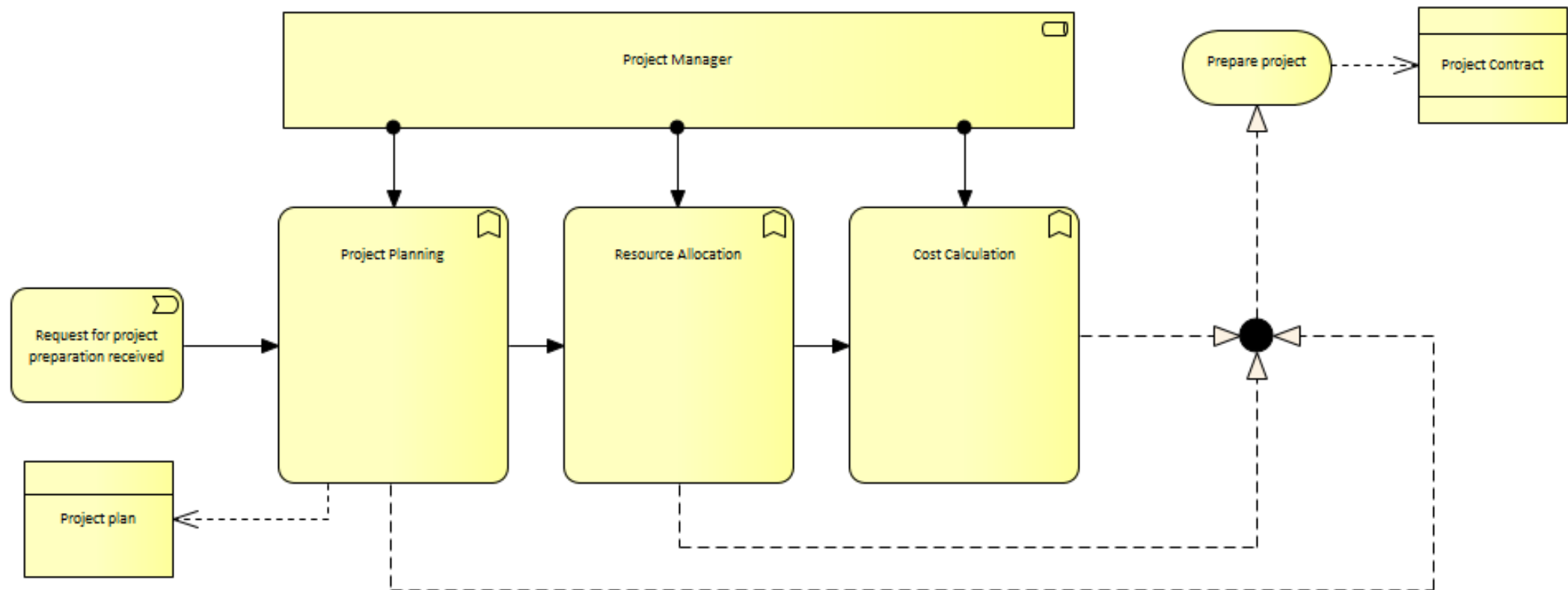
Definice :

- Stejně jako business proces, popisuje business funkce vnitřní chování (skryté před externími entitami) určité business role. Avšak zatímco proces je založen na sekvenci (Flow) aktivit, business funkce je založená na sdílených charakteristikách – zdrojích, kompetencích, business oblasti, znalostech apod.
- Další charakteristika business funkce (Business Function) je to, že přináší z pohledu businessu určitou přidanou hodnotu.
- Komplexní procesy zahrnují více aktivit, které mohou nabízet (být součástí) různých business funkcí. V tomto smyslu může být business proces vnímán jako řetězec business funkcí (viz obrázek).
- Business funkce může být spouštěna nebo spouštět (pomocí vazby triggering) jiné behaviorální business elementy (Business Event, Business Interaction, Business Process, Business Event nebo jinou Business function).
- Business funkce může být podporována (Served – vazbou Serving ve směru šipkou k procesu) jiným business procesem, business funkcí, business službou, Business interakcí a obdobnými elementy z aplikační a technologické vrstvy.
- Business role může být přiřazena (Assigned) business funkci.
- Business funkce může přistupovat k business objektům (a dalším Business Passive strukturám).
- Název Business Funkce by měl být definován tak, aby jasně vyjadřoval její chování: Customer Management, Request Administration, Payment Processing.

Elementy v Business Layer – Business Function #2

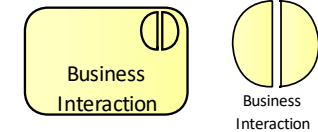


Elementy v Business Layer – Business Function #3



Elementy v Business Layer – Bus. Interaction #1

Typ: aktivní interní (Internal Behavior)



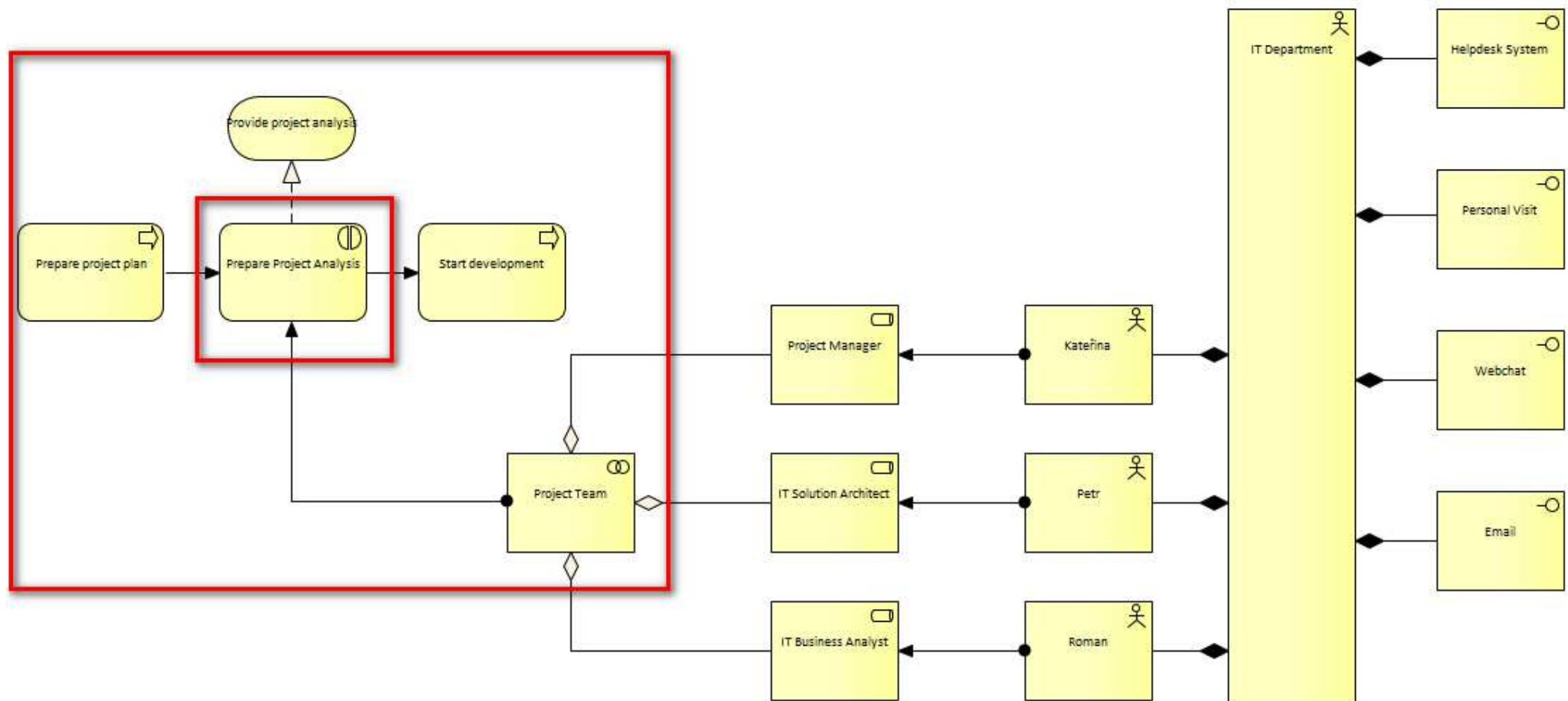
Definice:

Business interakce představuje společné chování, které je prováděno jednou či více business rolí (spojených do kolaborace – Business Collaboration).

Definice :

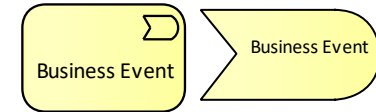
- Business Interaction se chová obdobně jako business proces nebo business funkce, ale zatím co proces či funkce jsou prováděny jednou rolí (a pokud je rolí na business proces nebo funkci připojeno více, pak mohou proces nebo funkci provádět nezávisle na sobě), business interakci může provádět pouze business kolaborace (Business Collaboration).
- Business Interaction může realizovat business službu.
- Business Interaction může být spouštěna nebo spouštět (pomocí vazby triggering) jiné behaviorální business elementy (Business Event, Business Function, Business Process, Business Event nebo jinou Business Interaction).
- Business Interaction může být podporována (Served – vazbou Serving ve směru šipkou k procesu) jiným business procesem, business funkcí, business službou, Business interakcí a obdobnými elementy z aplikační a technologické vrstvy.
- Pouze Business collaboration může být přiřazena (Assigned) business Interakci.
- Business Interaction může přistupovat k business objektům (a dalším Business Passive strukturám).
- Název Business Interaction by měl být definován tak, aby jasně vyjadřoval její smysl, preferováno je sloveso.

Elementy v Business Layer – Bus. Interaction #2



Elementy v Business Layer – Business Event #1

Typ: aktivní (Behavior)

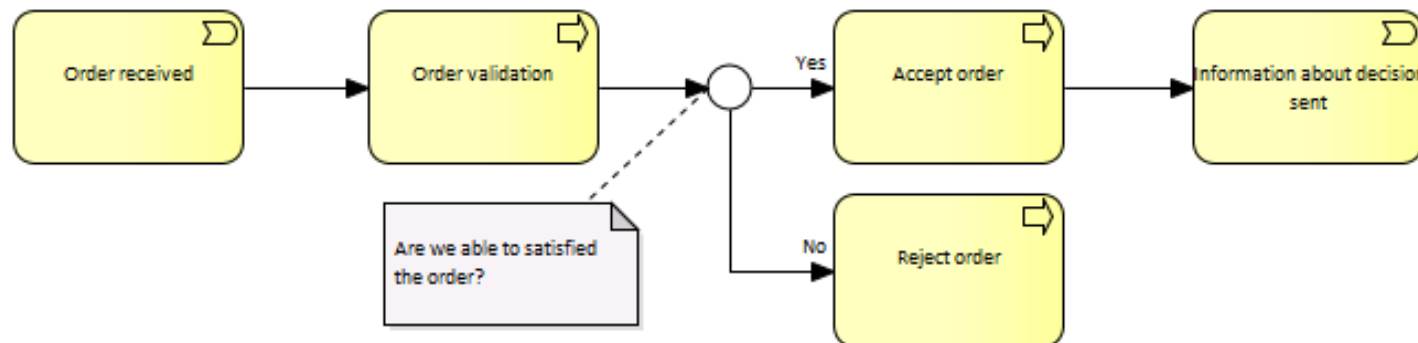


Definice:

Business event představuje změnu stavu. Může být vyvolána, či zachycena v rámci organizaci či mimo ní.

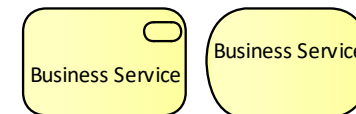
Definice :

- Na rozdíl od business procesů, funkcí a interakcí, business event nemá trvání – je okamžitý (nicméně, může mít časový údaj, kdy je event spouštěn – přesný čas, datum nebo interval, den v týdnu či měsíci, opakování).
- Business Process (a další business elementy chování – Business Function, Interaction) mohou být spouštěny a přerušovány elementem Business Event.
- Událost (Business Event) může nastat mimo organizaci (externím vlivem) nebo může být vyvolána business procesem (funkcí, interakcí...).
- Business event může být složen z více různých business events (událostí) pomocí vazby agregace nebo kompozice.
- V názvu by měla být činnost v dokonavém tvaru (objednávka přijata, dopis odeslán).



Elementy v Business Layer – Business Service #1

Typ: aktivní externí (Behavior)



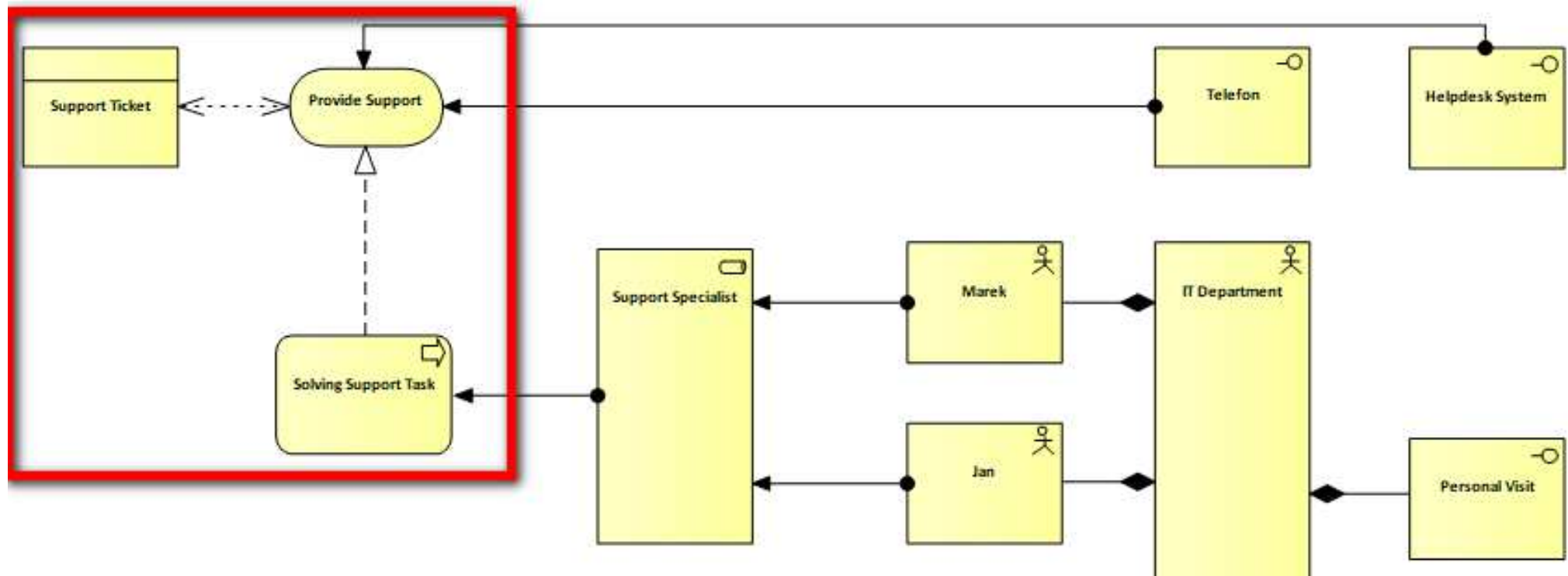
Definice:

Business služba (Business Service) reprezentuje explicitně definované službu či chování, vystavené do okolního prostředí.

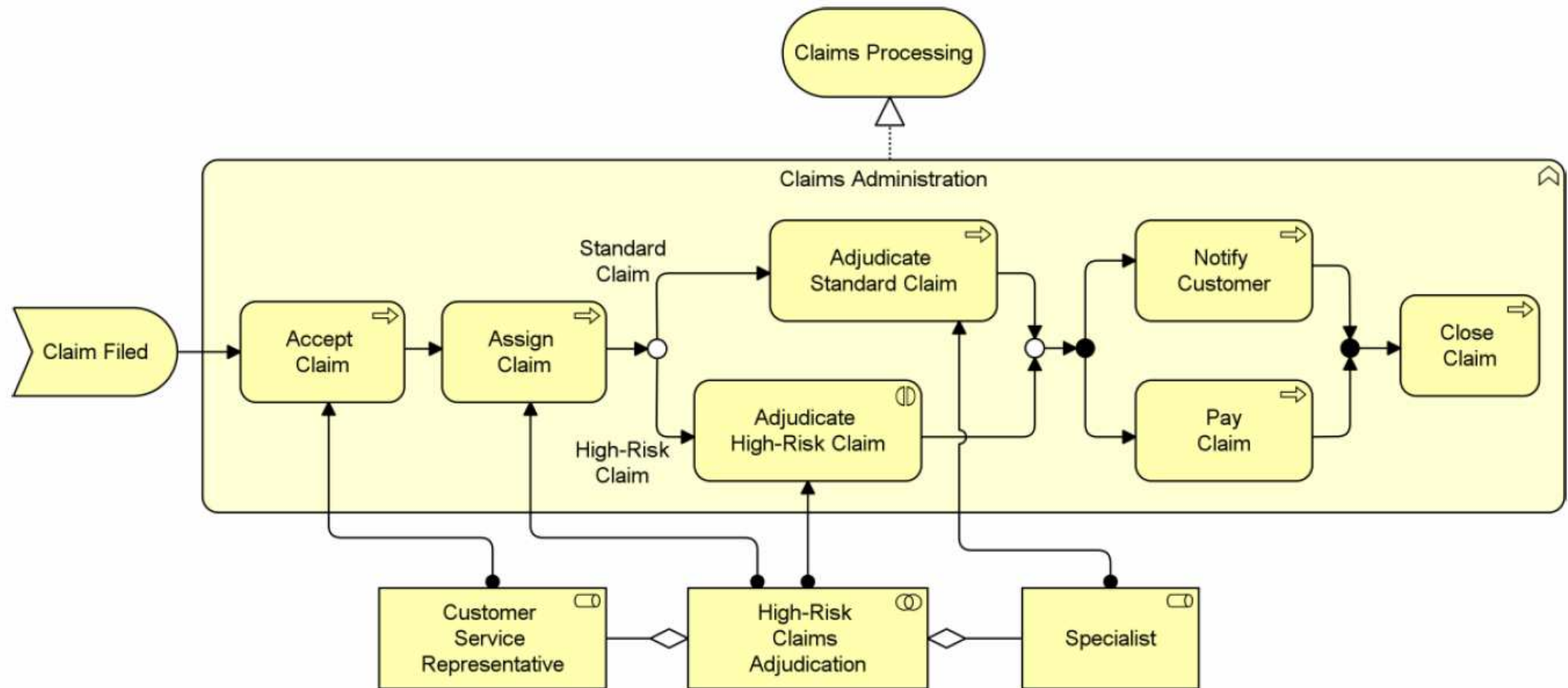
Definice :

- Business service vystavuje konkrétní funkcionalitu business rolí (Business Role) nebo business kolaborací (Business Collaboration) do okolního prostředí. Tato funkcionalita je dostupná prostřednictvím jednoho či více business rozhraní (Business Interface) – nepovinně.
- Business service může být realizována jiným business elementem, který představuje chování – Business Process, Function nebo Interaction.
- Business service může být externí (dostupná zákazníkům a jiným externím entitám) nebo interní, poskytující svou funkčnost dovnitř organizace.
- Business service může podporovat (vazba Serve po směru šipky) jiné business elementy chování (Behavior), Business Process, Function, Service nebo Interaction.
- Business Service může přistupovat k business objektům.

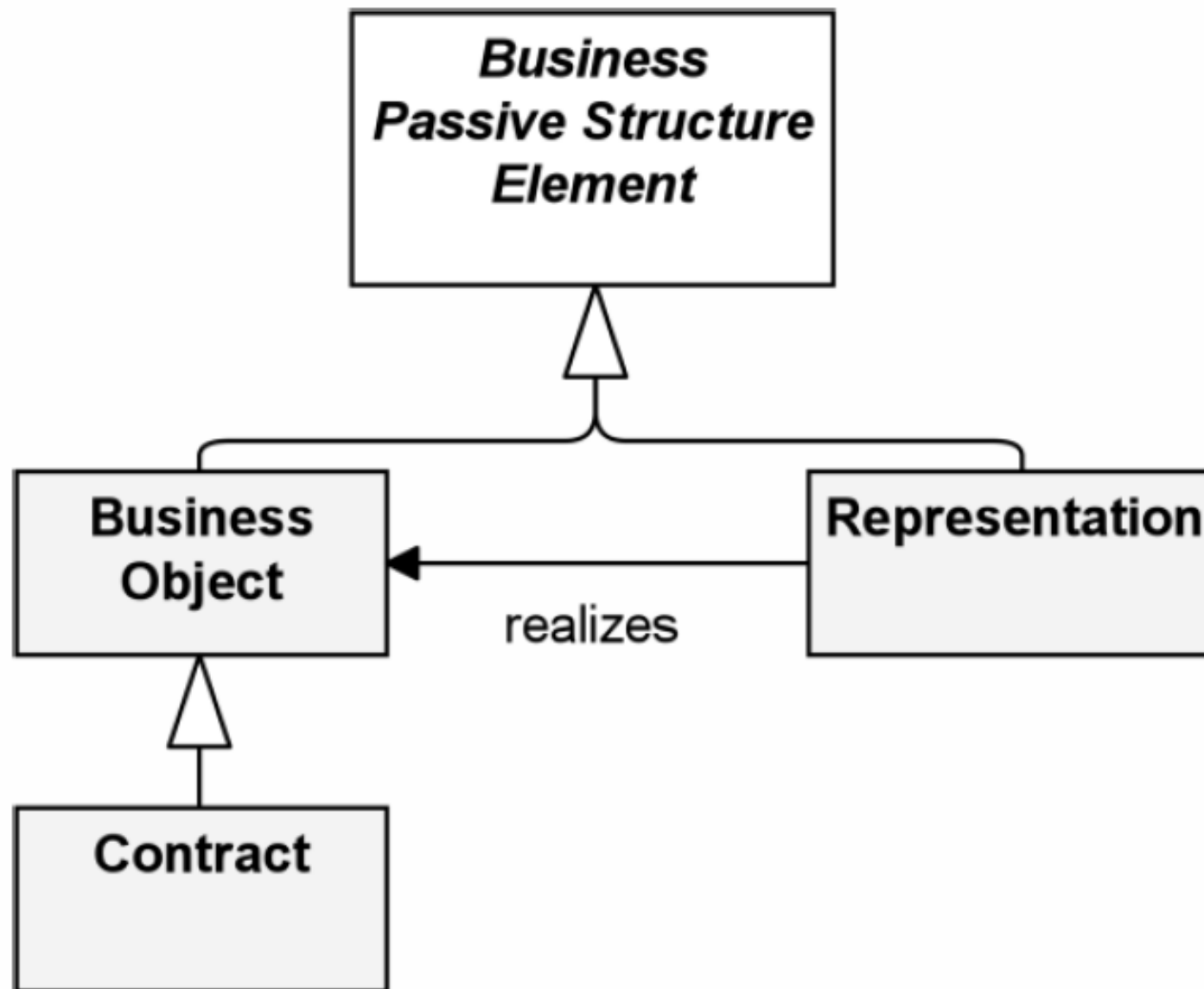
Elementy v Business Layer – Business Service #2



Elementy v Business Layer – Behavior Example

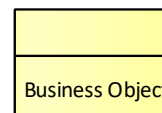


Elementy v Business Layer – Passive Structure



Elementy v Business Layer – Business Object #1

Typ: Passive Structure

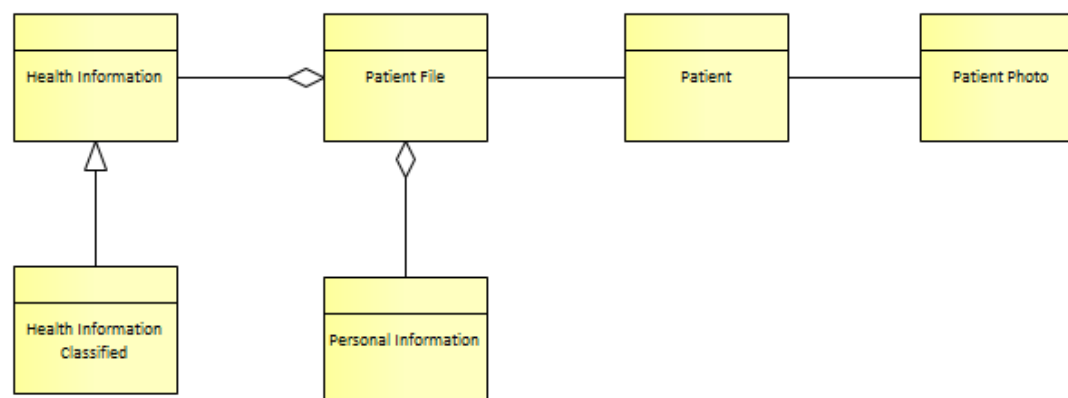
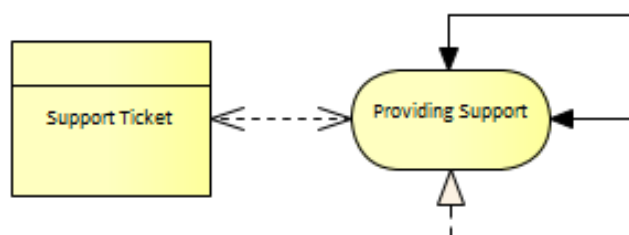


Definice:

Business objekt (Business object) reprezentuje koncept (Concept) použitý v rámci konkrétní business domény.

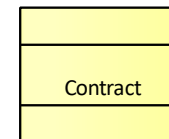
Definice :

- Business objekt (Business Object) reprezentuje data, informace či hmotné předměty (z business pohledu).
- Business objekty jsou pasivní, nemohou tedy spouštět (Trigger) business procesy, funkce apod.
- Business objekty odpovídají nejlépe třídám v jazyce UML (bez operací a atributů), pouze ve výjimečných případech jsou obdobou instancí.
- K business objektu mohou přistupovat business behaviorální elementy – Business Function, Process, Interaction, Service, Event. Přístupem se rozumí jejich vytváření, čtení, editace a rušení.
- Business objekty mohou být realizovány pomocí elementu Meaning a / nebo elementem Data Object (bude popsáno dále).
- Business objekty mohou být skládány z jiných business objektů pomocí agregací nebo kompozic, mohou mezi sebou být navázány pomocí asociací nebo mohou být specializovány (pomocí vazby specializace – Specialization.)
- Jméno business objektu by mělo obsahovat podstatné jméno.



Elementy v Business Layer – Business Contract #1

Typ: Passive Structure

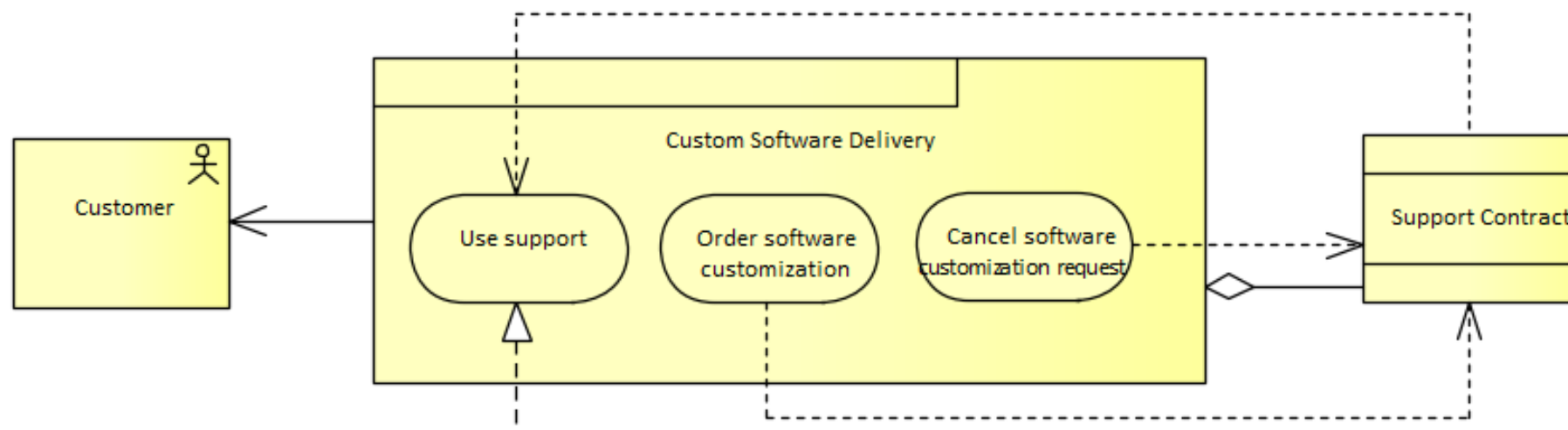


Define:

Business kontrakt (Business contract) formální či neformální popis dohody mezi poskytovatelem a konzumentem (služeb apod.), která obsahuje práva a povinnosti související s produktem. Dále může stanovit pravidla pro jejich komunikaci či jinou interakci.

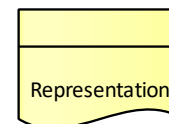
Define :

- Business kontrakt je specializací business objektu (tedy všechny vlastnosti a vztahy definované na úrovni business objektu platí i pro kontrakt).
- Business kontrakt (Business Contract) může být navíc agregován do produktu (Product), což business objekt nemůže.
- Business kontrakt může obsahovat například SLA (Service Level Agreement) a bližší specifikaci produktu.
- Název business kontraktu by měl obsahovat podstatné jméno.



Elementy v Bus. Layer – Bus. Representation #1

Typ: Passive Structure

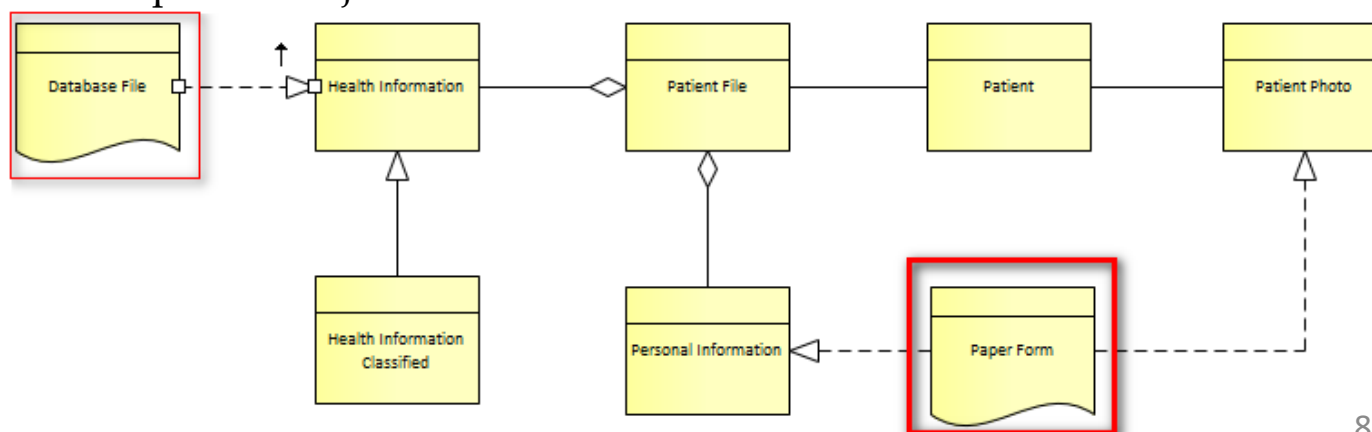


Definice:

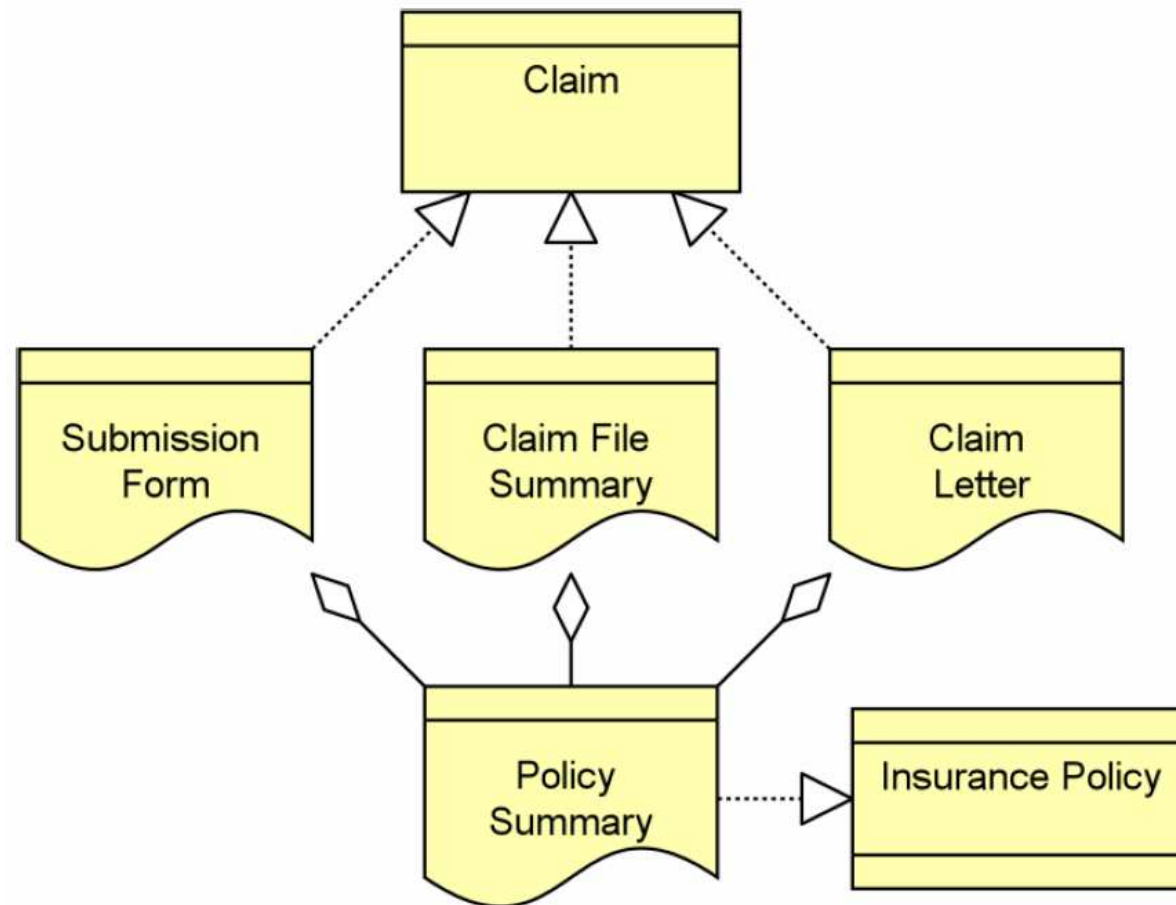
Business reprezentace (Representation) reprezentuje vnímatelnou (Perceptible) formu business objektů (Business Object). Reprezentace realizuje business objekty pomocí vazby Realization.

Definice :

- Business Representation jsou vnímatelné nosiče informací, které souvisejí s business objektem. Pokud je to v konkrétním případě relevantní, mohou být klasifikovány z hlediska různých kritérií, například dle média (audio, papírová forma, video, elektronická zpráva) nebo z hlediska formátu (HTML, RTF, PDF apod.).
- Jeden business objekt může mít více reprezentací a jedna reprezentace může realizovat více business objektů.
- Representation se mohou navzájem agregovat pomocí Aggregation nebo Composition.
- K Representation může být připojen Meaning (bude vysvětleno dále).
- Název Representation by měl obsahovat podstatné jméno.
- Representation může být aplikován na kontrakt.



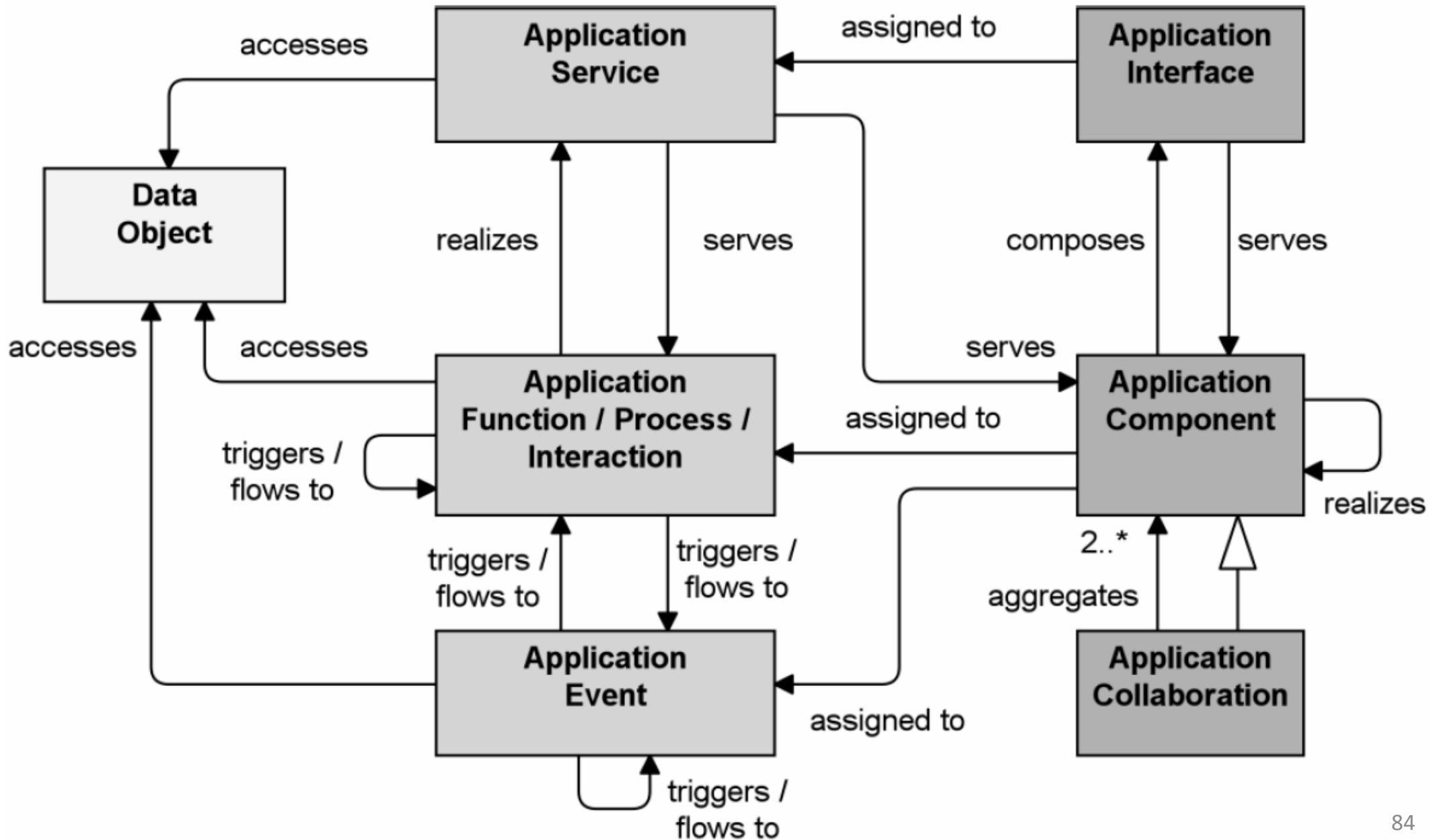
Elementy v Bus. Layer – Bus. Representation #2



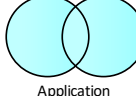
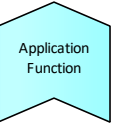
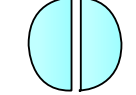
ArchiMate

Elementy – Application Layer

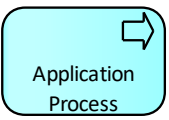
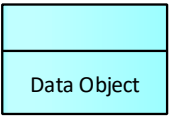
Application Layer - metamodel



Elementy v Application Layer – přehled #1

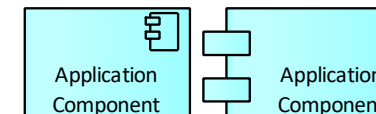
Element	Definice	Aspekt
 Application Component  Application Component	An application component represents an encapsulation of application functionality aligned to implementation structure, which is modular and replaceable. It encapsulates its behavior and data, exposes services, and makes them available through interfaces.	Active structure
 Application Interface  Application Interface	An application interface represents a point of access where application services are made available to a user, another application component, or a node.	Active structure
 Application Collaboration  Application Collaboration	An application collaboration represents an aggregate of two or more application components that work together to perform collective application behavior.	Active structure
 Application Function  Application Function	An application function represents automated behavior that can be performed by an application component.	Internal Behavior
 Application Service  Application Service	An application service represents an explicitly defined exposed application behavior.	External Behavior
 Application Interaction  Application Interaction	An application interaction represents a unit of collective application behavior performed by (a collaboration of) two or more application components.	Internal Behavior
 Application Event  Application Event	An application event is an application behavior element that denotes a state change.	Behavior

Elementy v Application Layer – přehled #2

Element	Definice	Aspekt
	An application process represents a sequence of application behaviors that achieves a specific outcome.	Internal Behavior
	A data object represents data structured for automated processing.	Passive Structure

Elementy v Application Layer – Component #1

Typ: Active Structure



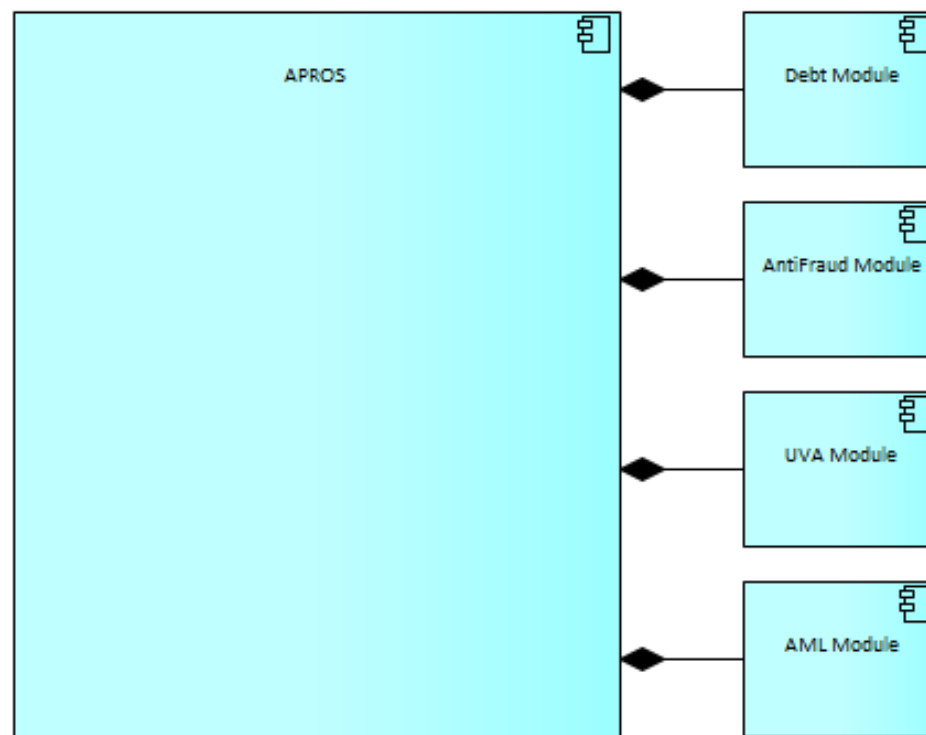
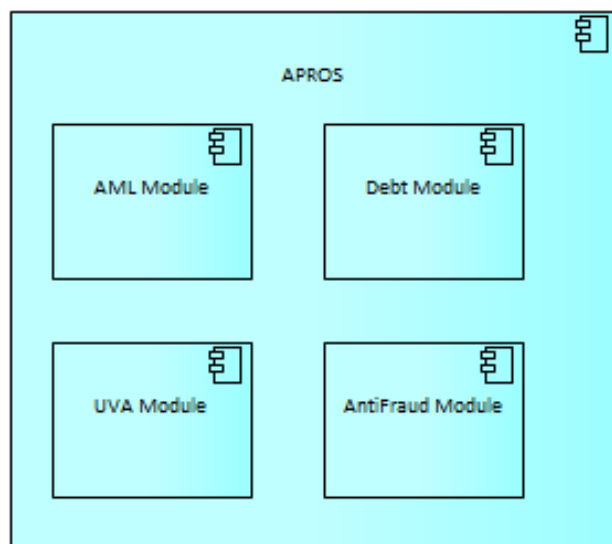
Definice:

Aplikační komponenta (Application Component) reprezentuje aplikační funkcionalitu (z pohledu implementace). Komponenta je modulární a nahraditelná. Zapouzdřuje své chování a data, vystavuje do okolního prostředí své služby prostřednictvím rozhraní (Application Interfaces).

Definice :

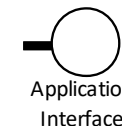
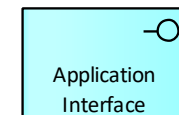
- Application Component často představují aplikace, systémy či subsystémy (z hlediska aplikační architektury a definice frameworku TOGAF).
- Application Component je vnímána jako samostatná jednotka a jako taková je nezávisle nasaditelná (deployable), znovupoužitelná a nahraditelná (replaceable).
- Application Component zapouzdřuje svůj obsah – svou funkcionalitu nabízí prostřednictvím souboru definovaných rozhraní (Application Interface).
- Application Component může být dekomponovaná do menších celků (subkomponent) pomocí kompozic (Composition) a agregací (Aggregation).
- Application Component mohou provádět společnou činnost (Behavior) pomocí Application Collaboration (bude vysvětleno dále).
- Application Component mohou být přiřazeny jedné či více aplikačním funkcím (Application Function).
- Aplikační rozhraní mohou podporovat (Serving) jiným aplikačním komponentám.
- Application Component podporují svou funkčností elementy business vrstvy (primárně behaviorální elementy).

Elementy v Application Layer – Component #2



Elementy v Application Layer – Interface #1

Typ: Active Structure



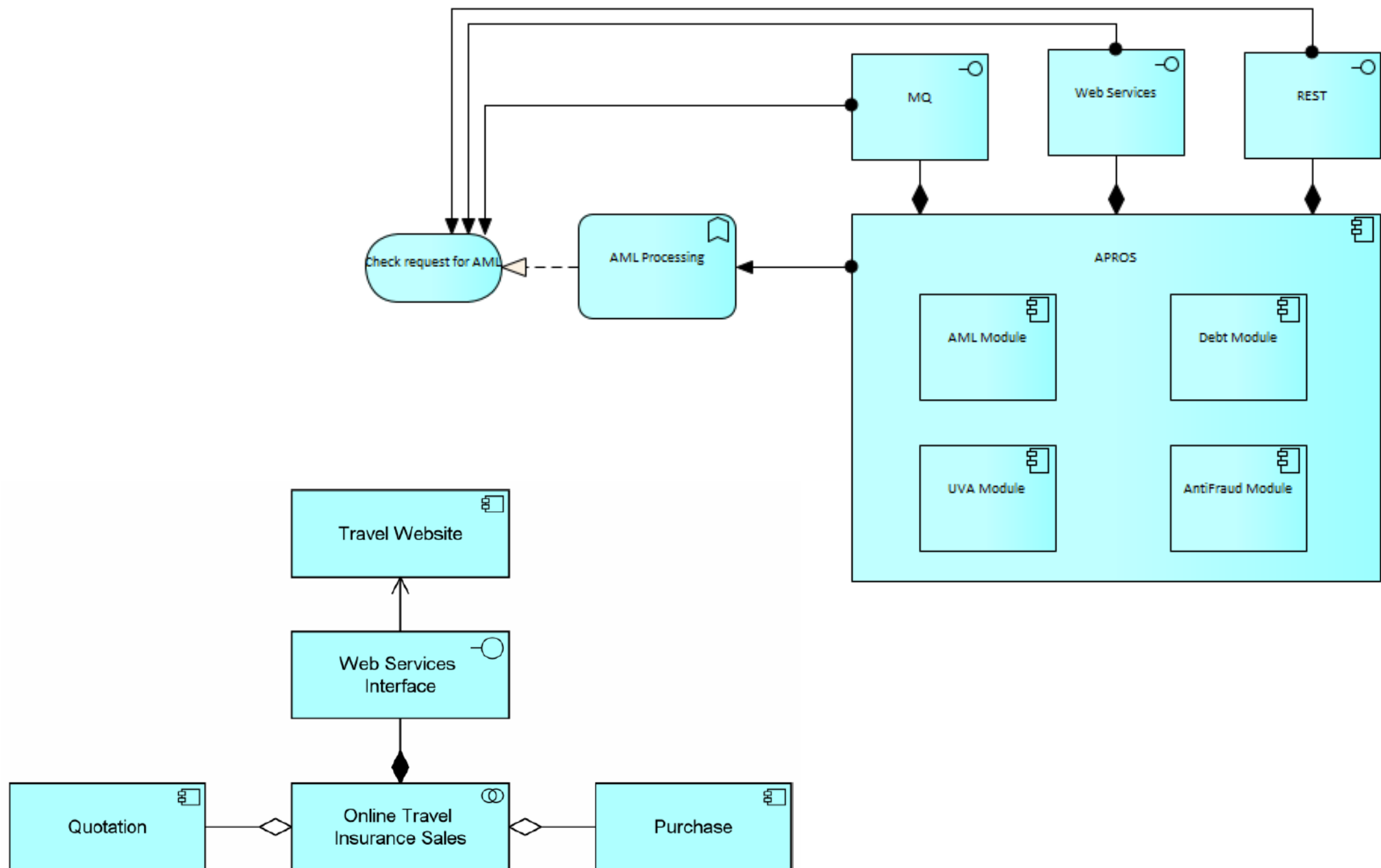
Definice:

Aplikační rozhraní (Application Interface) reprezentuje přístupový bod, kde je aplikační služba (Application Service) dostupná uživateli, jiné aplikační komponentě nebo technickému uzlu (Technology Node).

Definice :

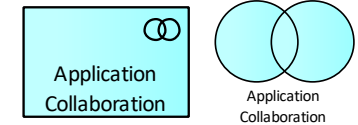
- Application Interface (aplikační rozhraní) specifikuje jakým způsobem je aplikační funkcionality dostupná jiným elementům.
- Application Interface vystavuje aplikační službu do okolního prostředí.
- Jeden Application Interface může vystavovat více aplikačních služeb (Application Service) a jedna aplikační služba může být vystavena pomocí více aplikačních rozhraní.
- Application Interface může být agregován do komponenty pomocí vazby Composition – to znamená, že prostřednictvím tohoto interface jsou dostupné všechny aplikačními služby (Application Service) a toto aplikační rozhraní může podporovat ostatní komponenty (pomocí vazby Serving přímo navázat na komponentu po směru šipky).
- Application Interface specifikuje komunikační protokoly, datové formáty, pre-condition, post-condition apod.
- Název Application Interface by měl obsahovat podstatné jméno.
- Application Interface se zpravidla pojmenovávají po technologiích.

Elementy v Application Layer – Interface #2



Elementy v Application Layer – Collaboration #1

Typ: Active Structure

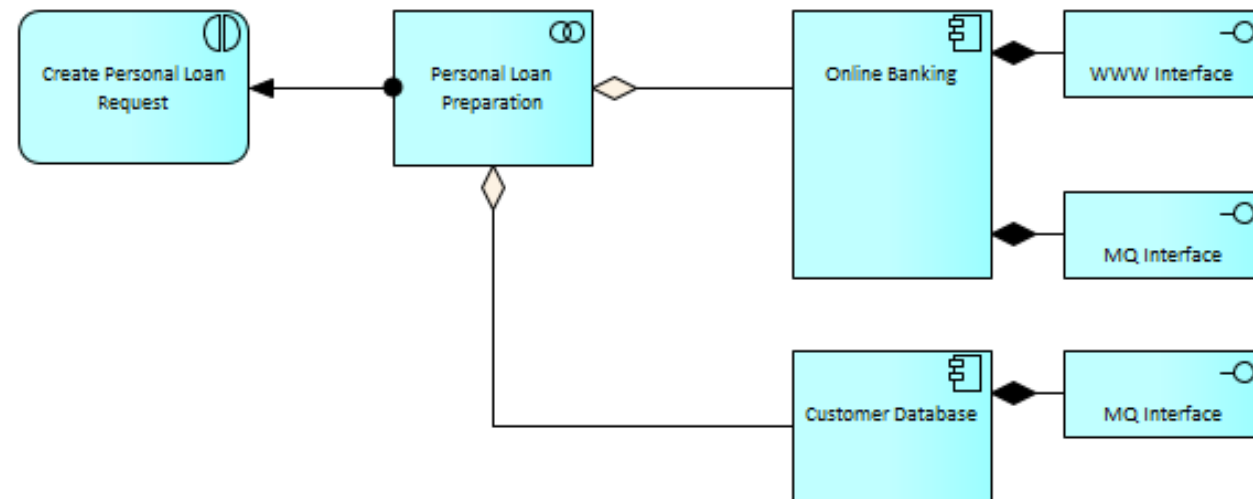


Definice:

Aplikační kolaborace (Application Collaboration) agreguje dvě nebo více aplikačních komponent (Application Component), které musí spolupracovat aby dosáhly společného účelu (společného chování – behavior).

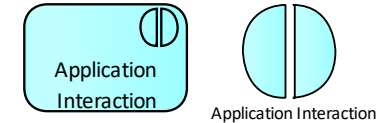
Definice :

- Application Collaboration definuje, které aplikační komponenty musí spolupracovat, aby splnily definovaný úkol. Tato spolupráce je modelována pomocí aplikační interakce (Application Interaction), na níž je Application Collaboration navázána pomocí vazby Assignment.
- Application Collaboration typicky modeluje dočasnou nebo i jen logickou spolupráci aplikačních komponent. Neexistuje tedy v aplikační architektuře jako samostatná komponenta.
- Application Collaboration může, podobně jako komponenta, podporovat (Serving po směru šipky) elementy (zpravidla Behavior) na Business Layer.
- Application Collaboration může být podporována (stejně jako komponenta) aplikačním rozhraním.
- Application Collaboration by měla mít v názvu podstatné jméno.



Elementy v Application Layer – Interaction #1

Typ: Behavior



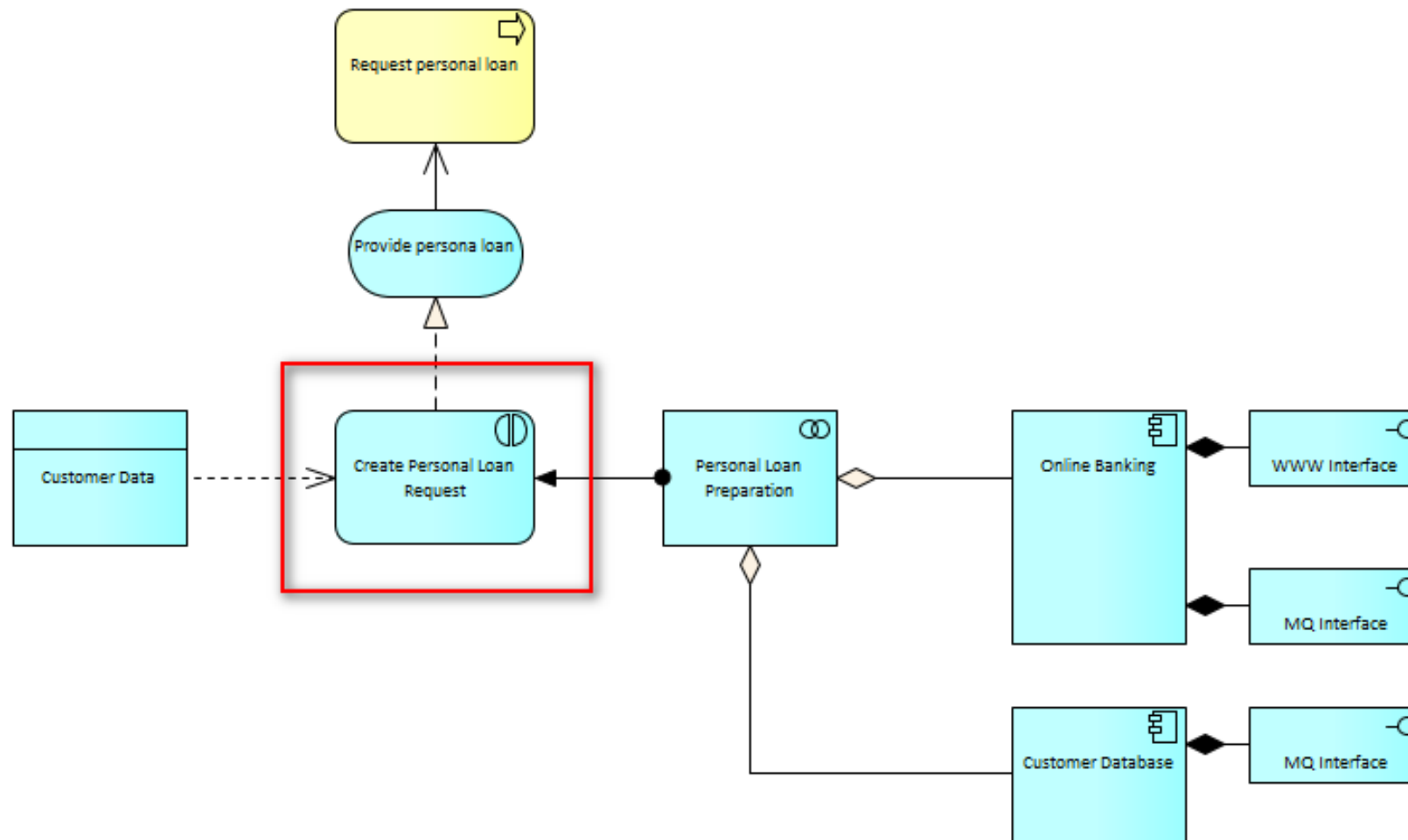
Definice:

Aplikační interakce reprezentuje konkrétní společné chování prováděné kolaborací (Application Collaboration) jedné či více aplikačních komponent.

Definice :

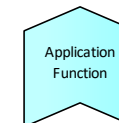
- Application Interaction popisuje společné chování aplikačních komponent, sdružených v aplikační kolaboraci (Application Collaboration). Interakce popisuje toto chování minimálně do té míry detailu, která je potřeba pro realizaci aplikační služby (Application Service). Kolaborace je na aplikační interakci navázána přes vazbu Assignment.
- Pro větší úroveň detailu lze Application Interaction popsat pomocí UML, například sekvenčními diagramy.
- Application Interaction může realizovat aplikační službu (Application Service).
- Aplikační a technologické služby (Technology Service) mohou podporovat Application Interaction (vazba Serving po směru šipky k Application Interaction).
- Application Interaction může přistupovat k datovým objektům (Data Object).
- Název Application Interaction by měl obsahovat sloveso a jasně vyjadřovat účel interakce.
- Application Interaction může vstupovat do všech standardních vztahů mezi aplikačními behaviorálními elementy, podporovat je, být podporována a interagovat mezi nimi pomocí vazby Triggering.

Elementy v Application Layer – Interaction #2



Elementy v Application Layer – Function #1

Typ: Internal Behavior



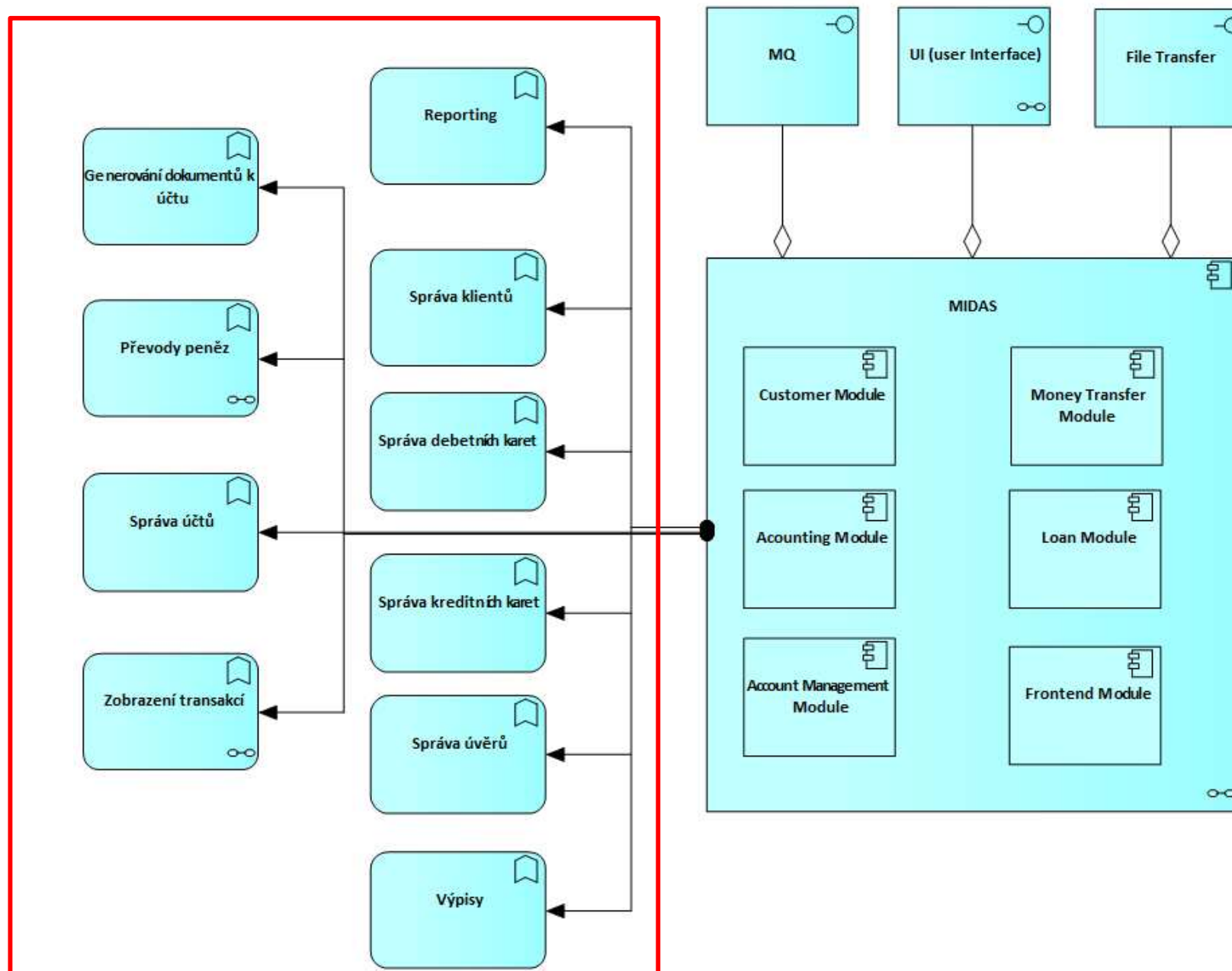
Definice:

Aplikační funkce (Application Function) reprezentuje automatické chování (činnost), které může být prováděno některou aplikační komponentou (Application Component).

Definice :

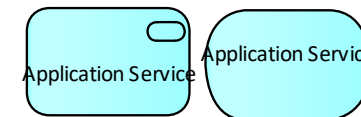
- Application Function popisuje interní chování nějaké aplikační komponenty. Pokud je toto chování vystaveno (exposed) do okolního prostředí, je tak učiněno prostřednictvím jedné či více aplikačních služeb (Application Service).
- Application Function abstrahuje od implementace (to je prováděno pomocí UML modelů), popisuje pouze nezbytné chování.
- Application Function může realizovat jednu či více aplikačních služeb.
- Application Function může být podporována (vazba Serving) aplikačními službami (Application Services) jiných Application Function.
- Application Function může přistupovat k datovým objektům.
- Aplikační komponenta může být přiřazena (vazba Assignment) aplikační funkci (Application Function), pak to znamená, že aplikační komponenta vykonává činnosti (behavior) popsané v Application Function.
- V názvu Application Function by mělo být sloveso a název by měl popisovat její činnost.

Elementy v Application Layer – Function #2



Elementy v Application Layer – Service #1

Typ: External Behavior



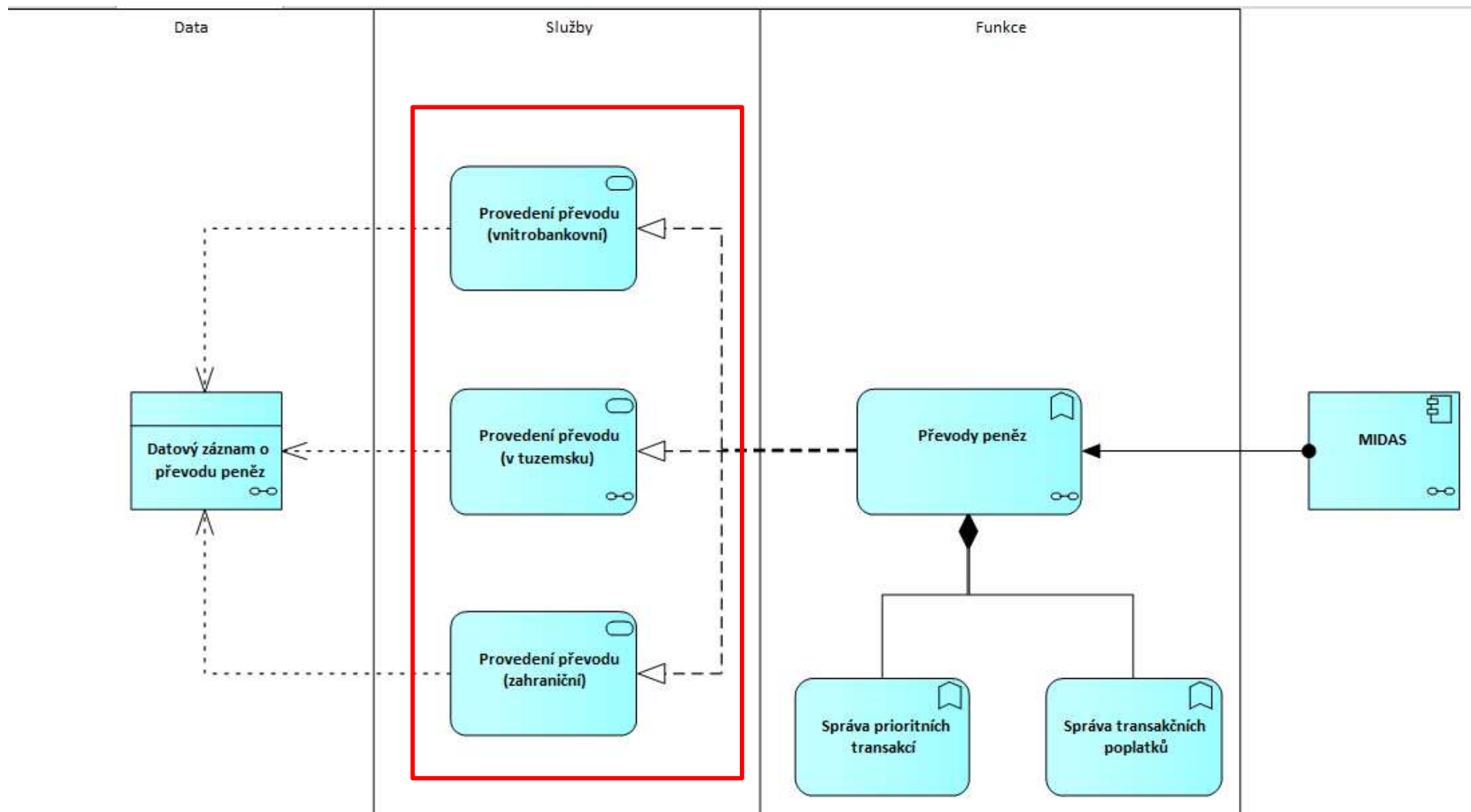
Definice:

Aplikační služba (Application Service) představuje explicitně definované chování (na aplikační úrovni), které je vystaveno do infrastruktury.

Definice :

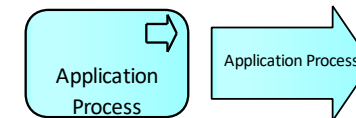
- Application Service vystavuje funkcionalitu komponent do prostředí. Tato funkcionalita je dostupná prostřednictvím jednoho či více Application Interface (vazbou Assignment) a jeden Application Interface může zpřístupňovat více Application Service.
- Application Service je realizována jednou či více aplikačních funkcí (přiřazených k aplikačním komponentám).
- Application Service může vytvářet, číst, editovat a mazat datové objekty (Data Object).
- Application Service by měla mít smysl z pohledu okolního prostředí a měla by poskytovat ucelenou a použitelnou službu svým uživatelům.
- Application Service může být chápána jako konkrétní API z pohledu integrační architektury.
- Aplikační službě (Application Service) může být přiřazena Value (viz další výklad).
- Application Service může podporovat (pomocí Serving směrem od Application Service) aplikační funkce, služby a behaviorální elementy z business vrstvy (business funkce, procesy, interakce, eventy).
- Název Application Service by měl obsahovat sloveso a popisovat konkrétní činnost. V AJ se doporučuje použít přímo slovo service.

Elementy v Application Layer – Service #2



Elementy v Application Layer – Process #1

Typ: Internal Behavior



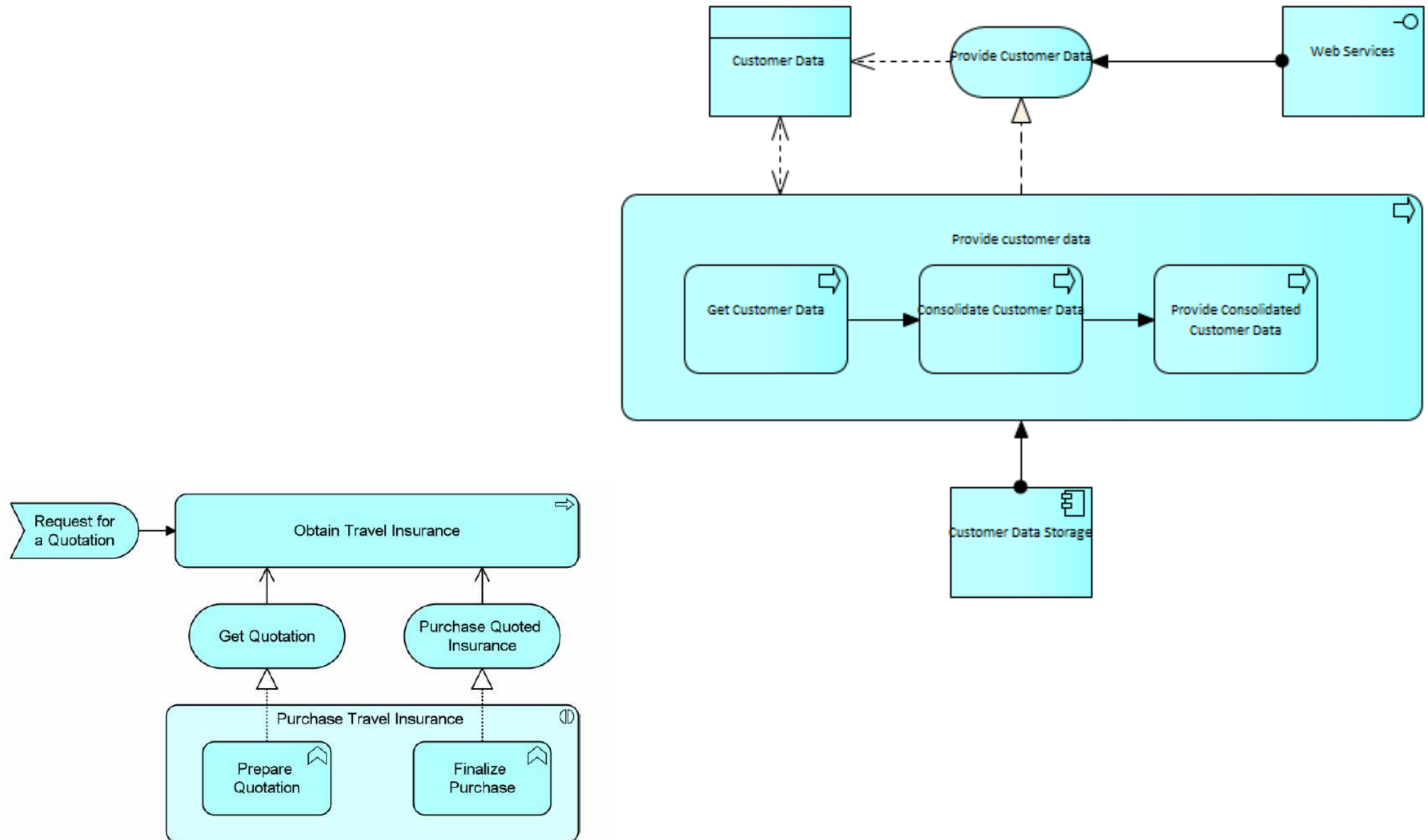
Definice:

Aplikační proces (Application Process) reprezentuje sekvenci akcí (či chování – Behavior) na aplikační úrovni, které musí být splněny pro dosažení požadovaného výstupu.

Definice :

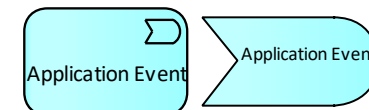
- Application Process popisuje interní chování aplikační komponenty, které je potřebné pro realizaci jedné či více aplikačních služeb (Application Service).
- Application Process je sekvence akcí vykonávaných aplikační komponentou v daném pořadí.
- Application Process může spouštět a být spouštěn jinými Application Process a dalšími aplikačními behaviorálními elementy (Application event, function, Interaction) pomocí vazby Triggering.
- Application Process může přistupovat k datovým objektům (vytvářet, číst, měnit a mazat).
- Aplikační komponenta může být přiřazena (Assignment) k Application Process, což znamená, že tato komponenta daný proces provádí.
- Application Process je z pohledu okolního prostředí blackbox, proto internal behavior.
- Application Process může realizovat jednu či více aplikačních služeb.
- Application Process může být podporován (vazba Serving) aplikačními službami.
- Application Process by měl obsahovat v názvu sloveso a popisovat účel procesu (Update Customer Data).

Elementy v Application Layer – Process #2



Elementy v App. Layer – Event #1

Typ: Behavior



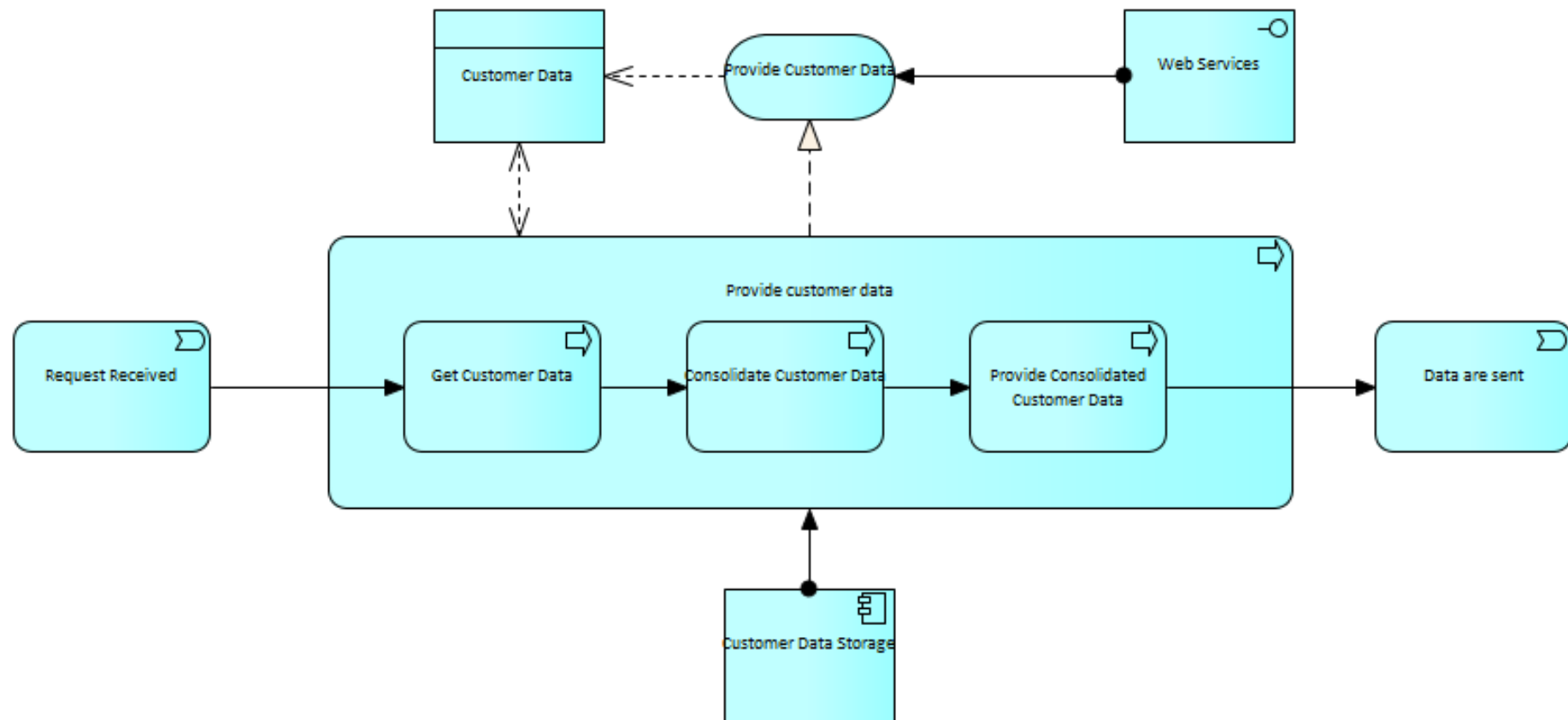
Definice:

Application Event je aplikační chování, které představuje změnu stavu.

Definice :

- Na rozdíl od aplikačních procesů, funkcí a interakcí, Application Event nemá trvání – je okamžitý (nicméně, může mít časový údaj, kdy je event spouštěn – přesný čas, datum nebo interval, den v týdnu či měsíci, opakování). Použitelné třeba pro „daily jobs“ apod.
- Application Process (a další aplikační elementy chování – Application Function, Interaction) mohou být spouštěny a přerušovány elementem Application Event.
- Application Event může nastat mimo organizaci (třeba externí aplikací) nebo může být vyvolána aplikačním procesem (funkcí, interakcí...).
- Application Event může být složen z více Application Event (událostí) pomocí vazby agregace nebo kompozice.
- V názvu by měla být činnost v dokonavém tvaru (žádost přijata).

Elementy v App. Layer – Event #2



Elementy v App. Layer – Data Object #1

Typ: Passive Structure

Data Object

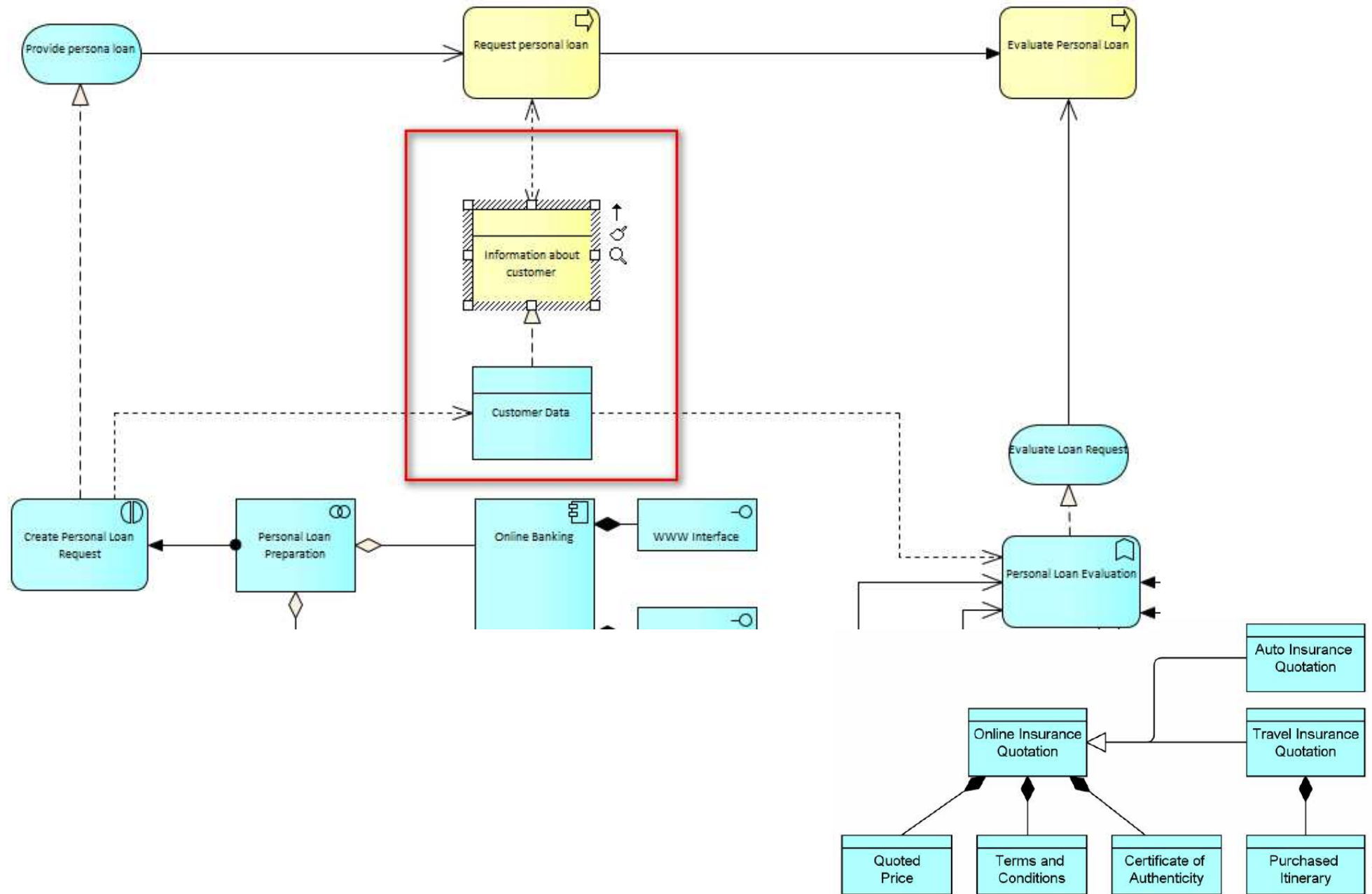
Definice:

Data Object reprezentuje strukturovaná data pro automatické zpracování.

Definice :

- Data Object reprezentuje data, informace či hmotné předměty s jasným významem i pro business, tedy nejen na aplikační úrovni. Příkladem může být záznam o zákazníkovi, databáze zákazníků apod.
- Data Object jsou pasivní, nemohou tedy spouštět (Trigger) aplikační procesy, funkce apod.
- Data Object odpovídají nejlépe třídám v jazyce UML (bez operací a atributů), pouze ve výjimečných případech jsou obdobou instancí.
- K Data Object mohou přistupovat aplikační behaviorální elementy – Application Function, Process, Interaction, Service, Event. Přístupem se rozumí jejich vytváření, čtení, editace a rušení.
- Data Object mohou být skládány z jiných Data Object pomocí agregací nebo kompozic, mohou mezi sebou být navázány pomocí asociací nebo mohou být specializovány (pomocí vazby specializace – Specialization.)
- Data Object mohou realizovat business objekty (a elementy z nich specializované).
- Data Object mohou být realizovány technologickými objekty a artefakty (Technology Object a Artifact na Technology Layer).
- Jméno Data Object by mělo obsahovat podstatné jméno.

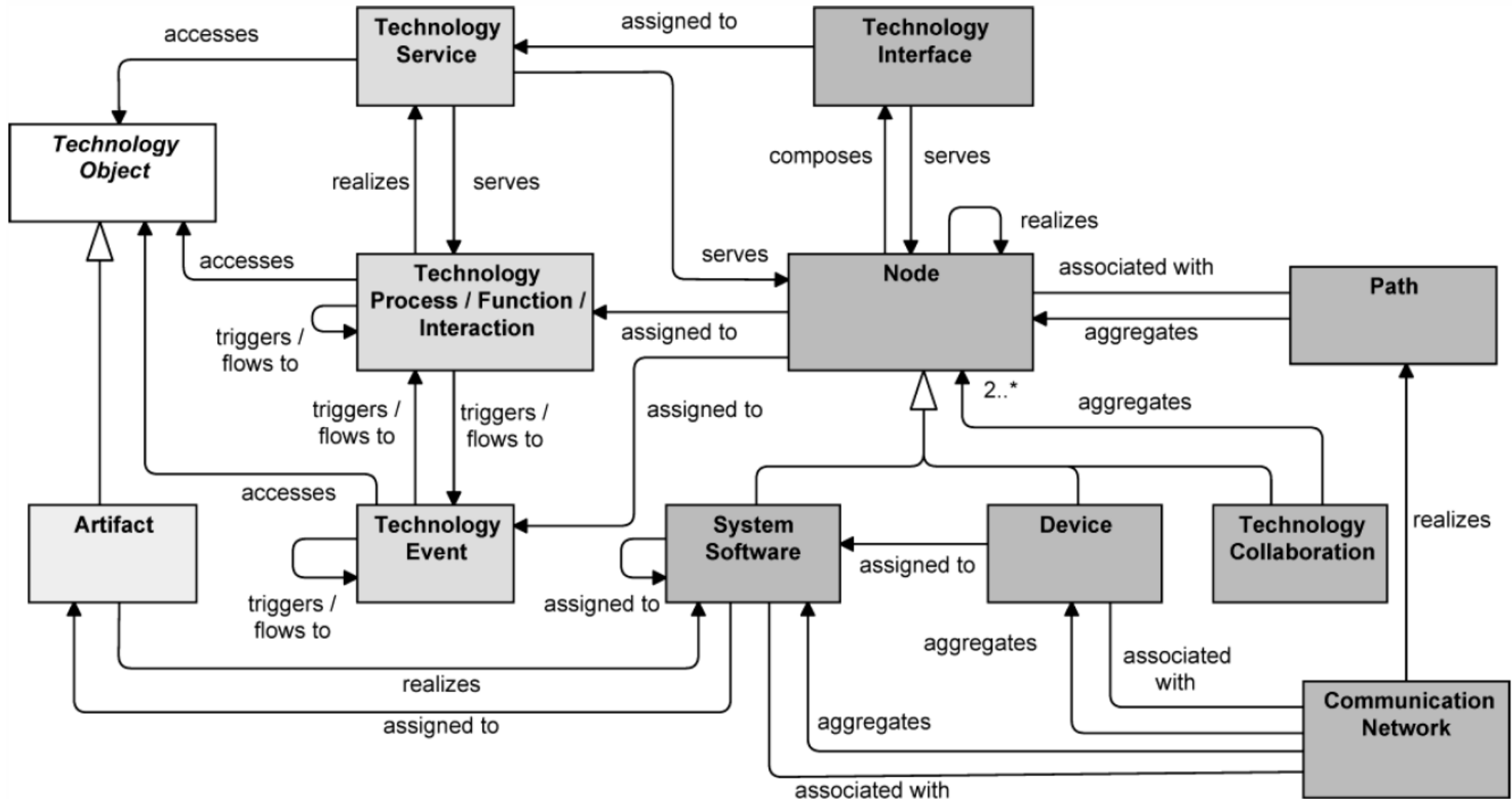
Elementy v App. Layer – Data Object #2



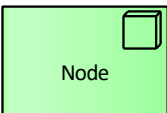
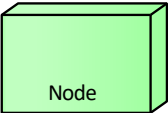
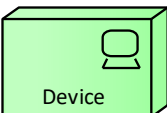
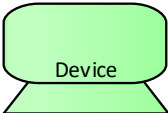
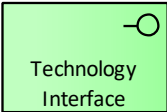
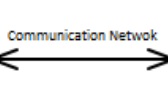
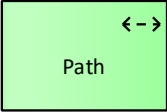
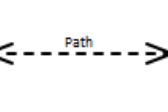
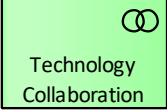
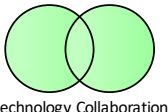
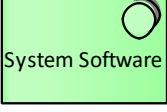
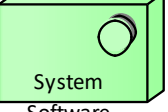
ArchiMate

Elementy – Technology Layer

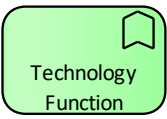
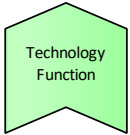
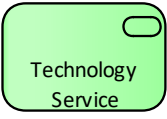
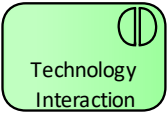
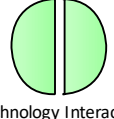
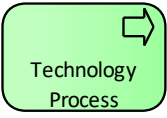
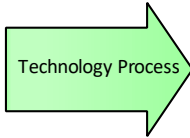
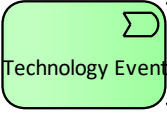
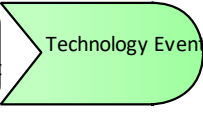
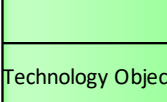
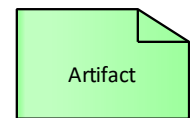
Technology Layer - metamodel



Elementy v Technology Layer – přehled #1

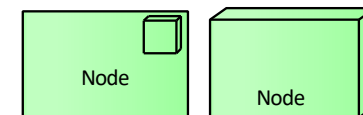
Element	Definice	Aspekt
 	A node represents a computational or physical resource that hosts, manipulates, or interacts with other computational or physical resources.	Active structure
 	A device is a physical IT resource upon which system software and artifacts may be stored or deployed for execution.	Active structure
 	A technology interface represents a point of access where technology services offered by a node can be accessed.	Active structure
 	A communication network represents a set of structures and behaviors that connects computer systems or other electronic devices for transmission, routing, and reception of data or data-based communications such as voice and video.	Active structure
 	A path represents a link between two or more nodes, through which these nodes can exchange data or material.	Active structure
 	A technology collaboration represents an aggregate of two or more nodes that work together to perform collective technology behavior.	Active structure
 	System software represents software that provides or contributes to an environment for storing, executing, and using software or data deployed within it.	Active structure

Elementy v Technology Layer – přehled #2

Element	Definice	Aspekt
 	A technology function represents a collection of technology behavior that can be performed by a node.	Internal Behavior
 	A technology service represents an explicitly defined exposed technology behavior.	External Behavior
 	A technology interaction represents a unit of collective technology behavior performed by (a collaboration of) two or more nodes.	Internal Behavior
 	A technology process represents a sequence of technology behaviors that achieves a specific outcome.	Internal Behavior
 	A technology event is a technology behavior element that denotes a state change.	Behavior
	A technology object represents a passive element that is used or produced by technology behavior.	Passive structure
 	An artifact represents a piece of data that is used or produced in a software development process, or by deployment and operation of an IT system.	Passive structure

Elementy v Tech. Layer – Node #1

Typ: Active Structure



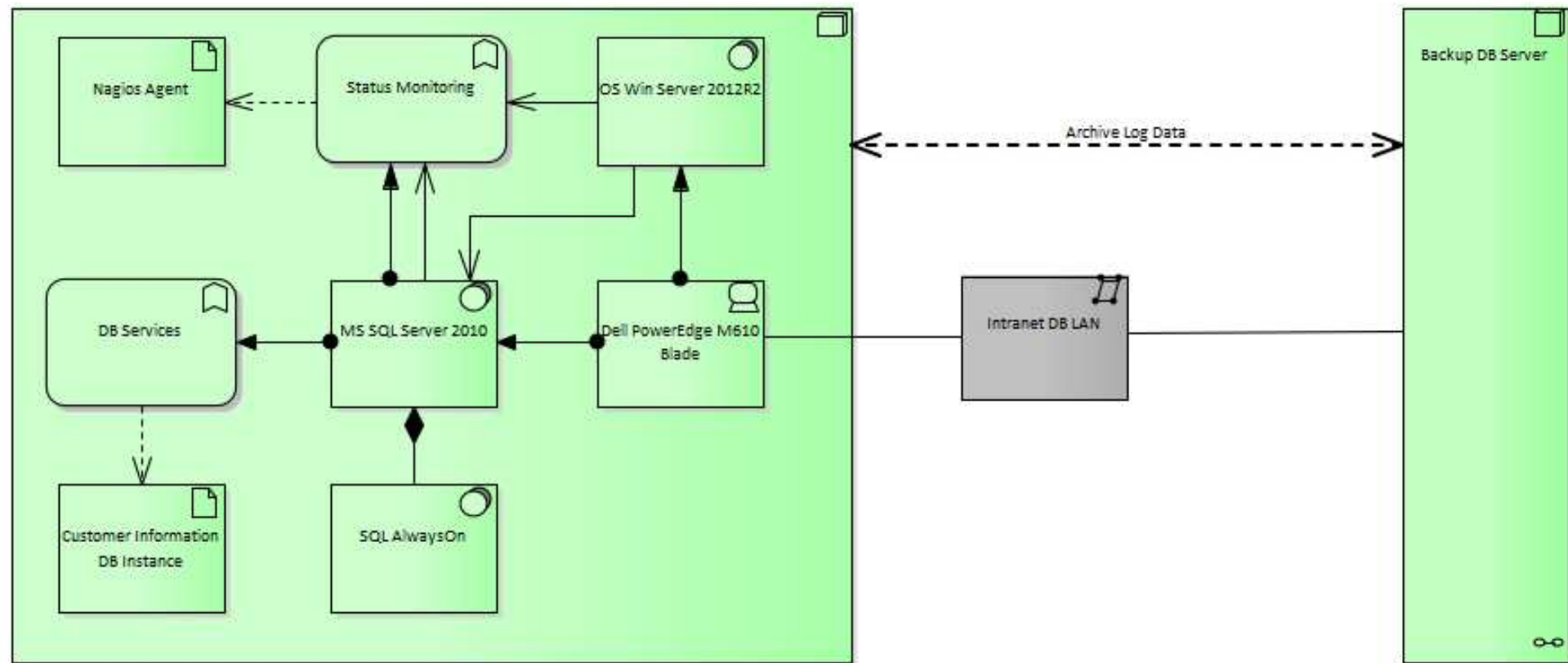
Definice:

Technology Node reprezentuje informační, výpočetní či fyzický zdroj, který je schopen hostovat, manipulovat nebo jinak interagovat s jinými informačními, výpočetními či fyzickými zdroji.

Definice :

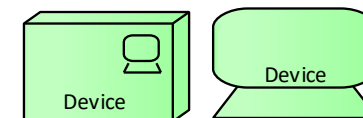
- Node představuje abstraktní entitu, která může obsahovat hardware (Device), systémový software (System Software), artefakty a může mít určité chování a mohou spouštět a manipulovat s technologickými artefakty (Technology Artifact) nebo s materiálem (Material z Physical Layer – bude popsáno dále).
- Node se dají agregovat pomocí agregace nebo kompozice.
- Node může reprezentovat například aplikační nebo databázový server (či jinou aplikační platformu) nebo celé prostředí (z pohledu fyzické architektury).
- Node z pohledu TOGAF: Soubor hardwarových a softwarových komponent, které poskytují služby pro podporu aplikací.
- Nody mohou být navzájem propojené pomocí vazby Path. Jedná se o abstraktní spojení (fyzické probíhá na úrovni HW a používá se k tomu element / vazba Communication Network).
- Artefakty mohou být modelovány přímo do Node nebo přiřazeny pomocí Assignment.
- Název Node by měl obsahovat podstatné jméno.

Elementy v Tech. Layer – Node #2



Elementy v Tech. Layer – Device #1

Typ: Active Structure



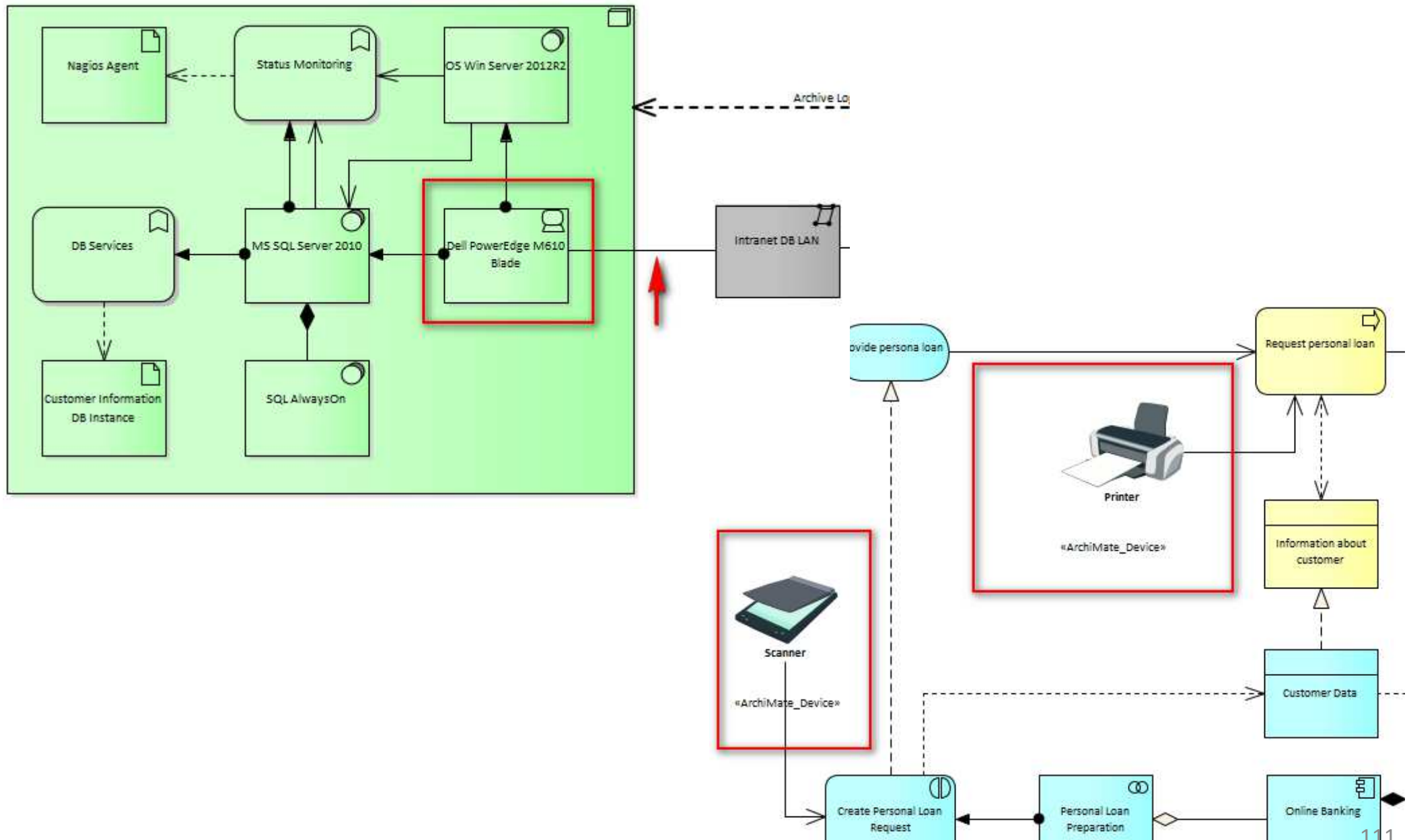
Definice:

Device (zařízení či hardware) reprezentuje fyzický IT zdroj, který je schopný spouštět a uchovávat (store) systémový software (System Software) či technologické artefakty (které reprezentují aplikační software).

Definice :

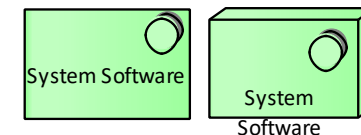
- Device je specializací Node a reprezentuje fyzický hardware.
- Device se dají navzájem agregovat pomocí agregace nebo kompozice.
- Device může reprezentovat PC, mainframe, tablet, tiskárnu, scanner nebo třeba HW firewall.
- Device může být propojeno pomocí sítí (Communication Network).
- Device bývá součástí node (buď modelován přímo do Node, nebo napojen na něj pomocí kompozice či agregace).
- Doporučuje se používat různé ikonky pro znázornění typu device – rozlišení mainframe od PC či telefonu.
Poznámka: toto doporučení není příliš časté a je to velmi užitečné.
- Device může být přiřazeno (pomocí vazby Assigned) k artefaktům (Technology Artifact) nebo systémovému software (System Software) pro znázornění, že jsou tyto prvky nasazeny (deployed) na tomto zařízení (Device).
- Název Device by měl obsahovat podstatné jméno a měl by se odkazovat na příslušný HW (např. Dell PowerEdge M610 Blade).

Elementy v Tech. Layer – Device #2



Elementy v Tech. Layer – System Software #1

Typ: Active Structure

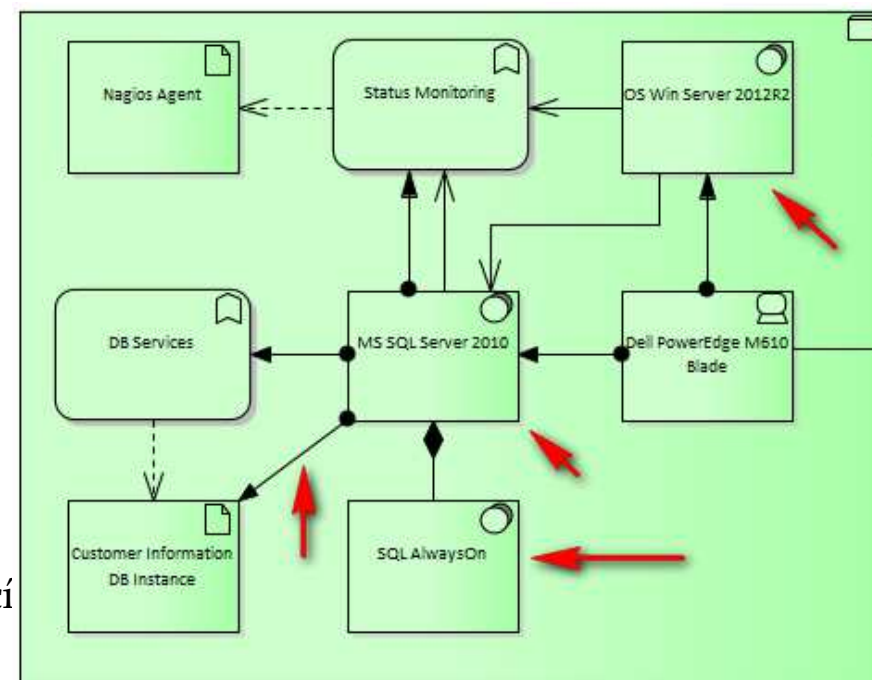


Definice:

System Software reprezentuje software, které poskytují prostředí (nebo jej spoluvytvářejí) pro ukládání, spouštění a používání aplikačního software nebo dat.

Definice :

- System Software je specializací Node a je používán pro modelování softwarového prostředí, ve kterém mohou být spouštěny artefakty.
- System Software se dají agregovat pomocí agregace nebo kompozice.
- System Software může být přiřazen k jinému System Software.
- System Software obvykle reprezentuje operační systém, aplikační kontejnery (JRE), databázový engine, monitorovací sw apod.
- System Software může být přiřazen (Assignment) k artefaktu pro znázornění, že tento artefakt běží v jeho prostředí.
- Device může být přiřazeno k System Software pro znázornění, že tento software je na něm nasazen (a běží v jeho prostředí).
- Název System Software by mělo být podstatné jméno, které vyjadřuje jeho charakteristiku (například MySQL 5.3).



Elementy v Tech. Layer – Tech. Collaboration #1

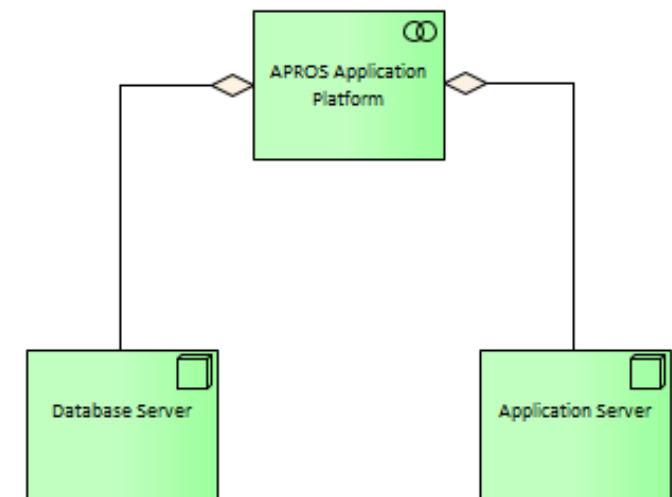
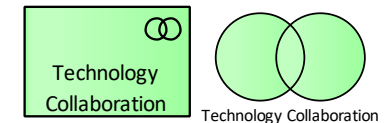
Typ: Active Structure

Definice:

Technologická kolaborace (Application Collaboration) agreguje dva nebo více nodů (Technology Node), které musí spolupracovat aby dosáhly společného účelu (společného chování – behavior).

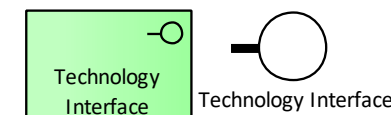
Definice :

- Technology Collaboration definuje, které Node (uzle) musí spolupracovat, aby splnily definovaný úkol. Tato spolupráce je modelována pomocí technologické interakce (Technology Interaction), na níž je Technology Collaboration navázána pomocí vazby Assignment. Může být navázána i na jiné behaviorální elementy z Technology Layer (Technology process, function...).
- Technology Collaboration typicky modeluje dočasnou nebo i jen logickou spolupráci více Node. Neexistuje tedy v aplikační architektuře jako samostatný Node.
- Technology Collaboration může, podobně jako Node, podporovat (Serving po směru šipky) elementy (zpravidla Behavior) na Application Layer.
- Technology Collaboration je specializací Node.
- Technology Collaboration může být podporována (stejně jako Node) technologickým rozhraním (Technology Node).
- Technology Collaboration by měla mít v názvu podstatné jméno a název by měl vyjadřovat účel kolaborace.



Elementy v Tech. Layer – Technology Interface #1

Typ: Active Structure

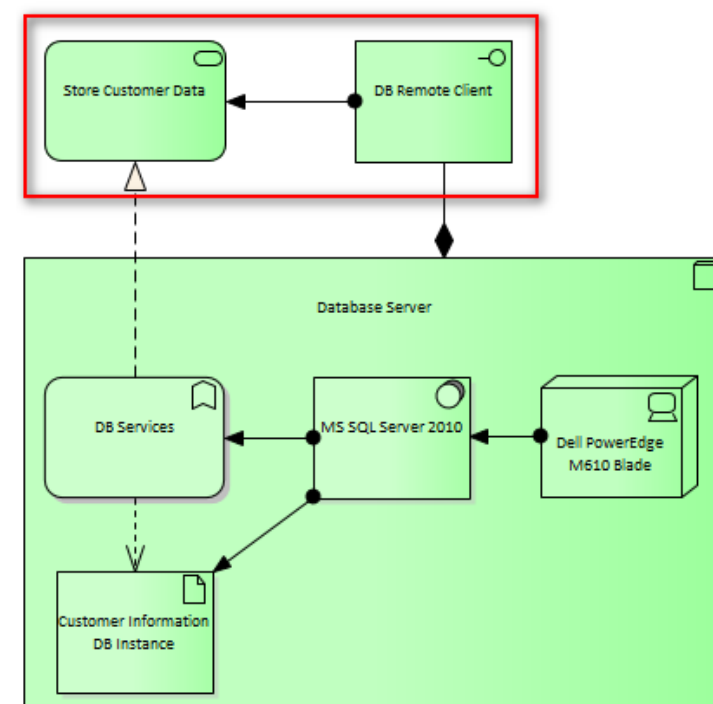


Define:

Technologické rozhraní (Technology Interface) reprezentuje přístupový bod, kde je technologická služba (Technology Service) nabízená Technology Node dostupná.

Define :

- Technology Interface (technologické rozhraní) specifikuje, jakým způsobem je Technology Service dostupná jiným technologickým uzlům (Technology Nodes).
- Technology Interface vystavuje technologickou službu do okolního prostředí.
- Jeden Technology Interface může vystavovat více technologických služeb (Technology Service) a jedna aplikační služba může být vystavena pomocí více technologických rozhraní.
- Technology Interface může být agregován do Node pomocí vazby Composition – to znamená, že prostřednictvím tohoto interface jsou dostupné všechny technologické služby (Technology Service) a toto aplikační rozhraní může podporovat ostatní Technology Node (pomocí vazby Serving přímo navázat na Node po směru šipky).
- Technology Interface specifikuje komunikační protokoly, datové formáty, pre-condition, post-condition apod.
- Název Technology Interface by měl obsahovat podstatné jméno.



Elementy v Tech. Layer – Path #1

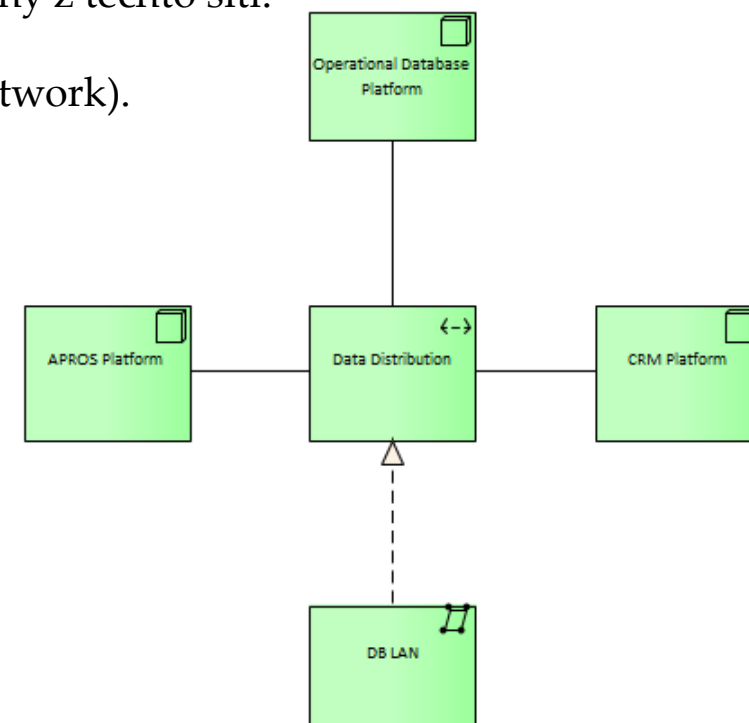
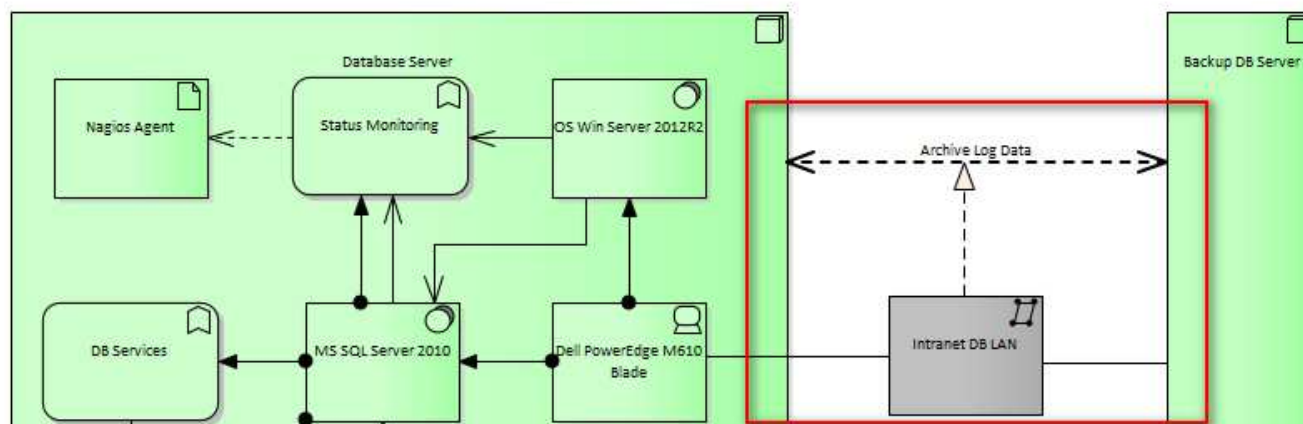
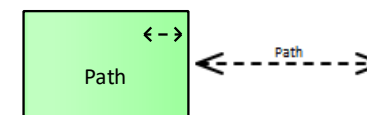
Typ: Active Structure

Definice:

Path reprezentuje spojení (na logické úrovni) dvou a více Nodes, skrze které si tyto Nodes mohou vyměňovat data nebo materiál (Material).

Definice :

- Path se používá pro modelování logické komunikace (nebo distribuce) mezi uzly (Technology Node). Tato logická komunikace je na fyzické úrovni realizována pomocí jedné či více sítí (Communication Network). Parametry Path (bandwidth, latence apod.) jsou zpravidla agregovány z těchto sítí.
- Path propojuje dva a více Node.
- Path může být realizován jednou či více sítěmi (Communication Network).
- Path může agregovat Nody.



Elementy v Tech. Layer – Communication Network #1

Typ: Active Structure

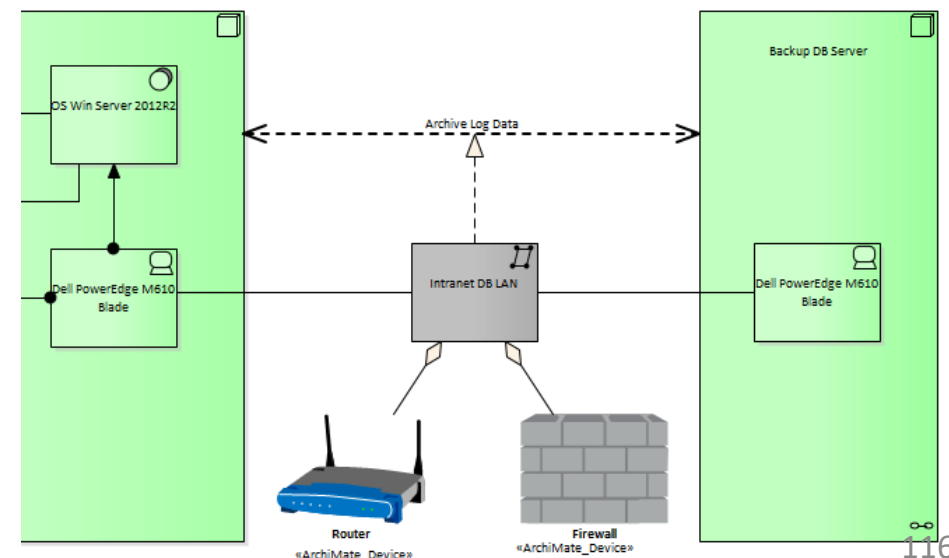


Definice:

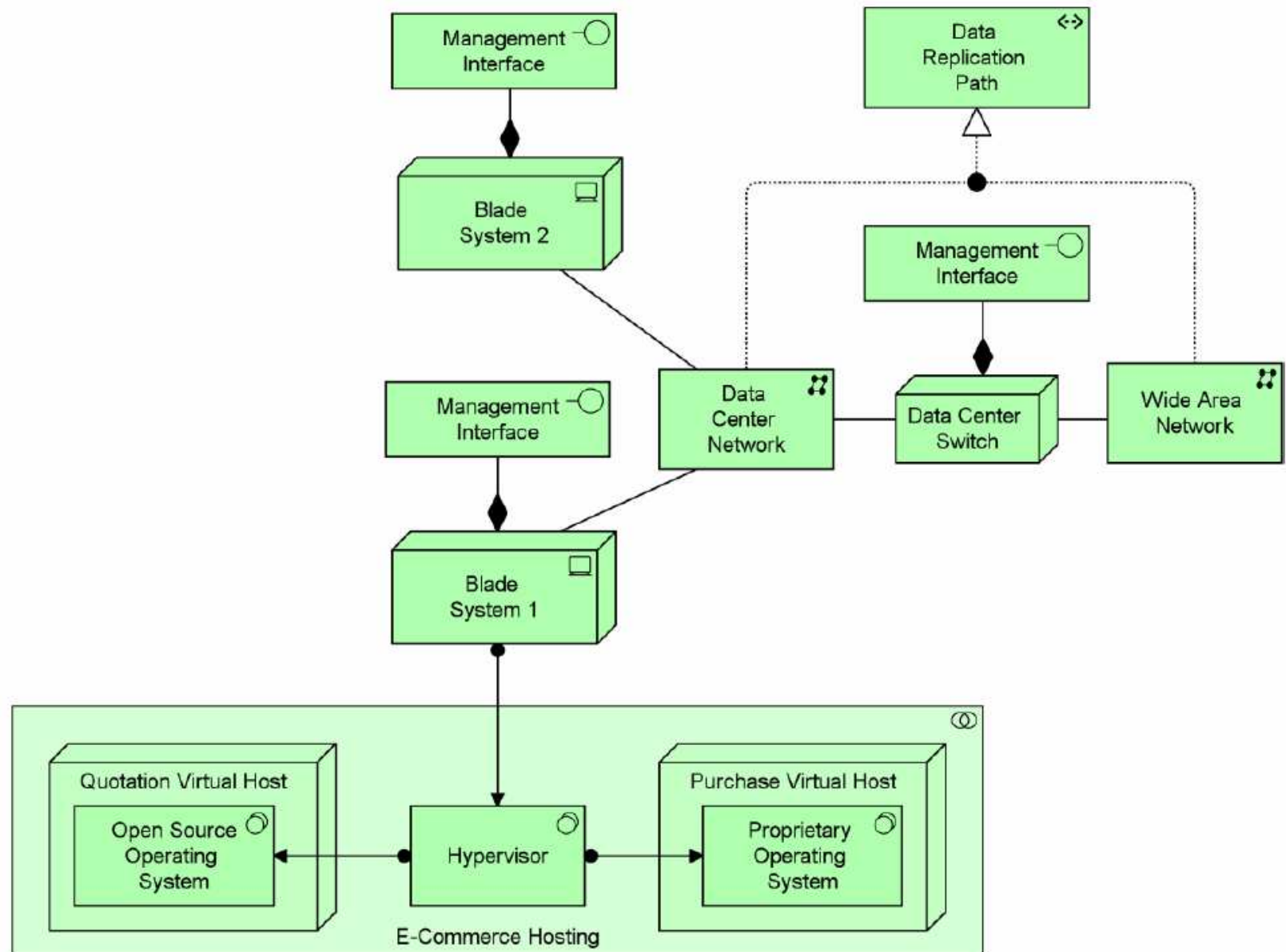
Communication Network reprezentuje soubor technických prostředků a jejich chování, které je potřeba k propojení počítačů, počítačových systémů či jiných elektronických zařízení. Toto propojení je schopné odesílat, routovat a přijímat data nebo jinou komunikaci založenou na datech, jako je například hlas či video.

Definice :

- Communication Network propojuje dvě a více zařízení. Nejjednodušší varianta takové komunikace je jedna prostá linka (kabel) mezi dvěma zařízeními. Složitější komunikace představuje mnoho takových linek a dodatečných síťových zařízení (router, firewall, switche apod.).
- Communication Network se může sestávat ze subsítí (Communication Network).
- Communication Network může (ve formě elementu) agregovat zařízení (Devices), systémový software (Systém Software), například firewally, routery, switche.

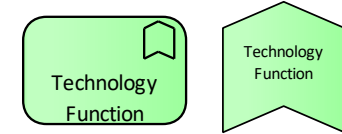


Elementy v Tech. Layer – Communication Network #2



Elementy v Technology Layer – Technology Function #1

Typ: Internal Behavior



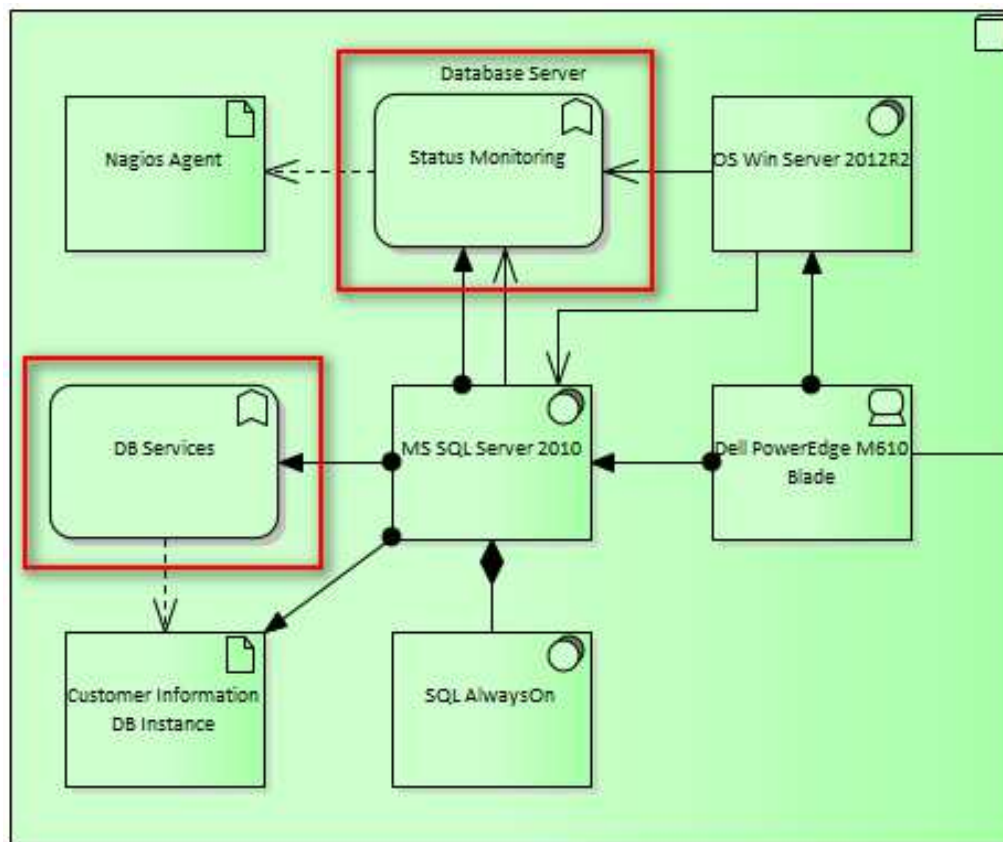
Definice:

Technologická funkce (Technology Function) reprezentuje automatické chování (činnost), které může být prováděna některým (Technology) Node.

Definice :

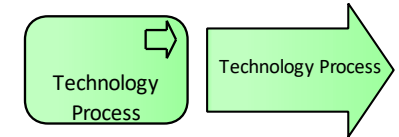
- Technology Function popisuje interní chování konkrétního Node (a jeho specializací, tedy například Device nebo System Software). Pokud je toto chování vystaveno (exposed) do okolního prostředí, je tak učiněno prostřednictvím jedné či více technologických služeb (Technology Service).
- Technology Function abstrahuje od implementace (to je prováděno pomocí UML modelů), popisuje pouze nezbytné chování.
- Technology Function může realizovat jednu či více technologických služeb.
- Technology Function může být podporována (vazba Serving) technologickými službami (Technology Services) jiných Technology Function.
- Technology Function může přistupovat k technologickým objektům (Technology Object) a artefaktům.
- Node může být přiřazen (vazba Assignment) technologické funkci (Technology Function), pak to znamená, že navázaný Node vykonává činnosti (behavior) popsané v Technology Function.
- V názvu Technology Function by mělo být sloveso a název by měl popisovat její činnost.

Elementy v Technology Layer – Technology Function #2



Elementy v Technology Layer – Technology Process #1

Typ: Internal Behavior



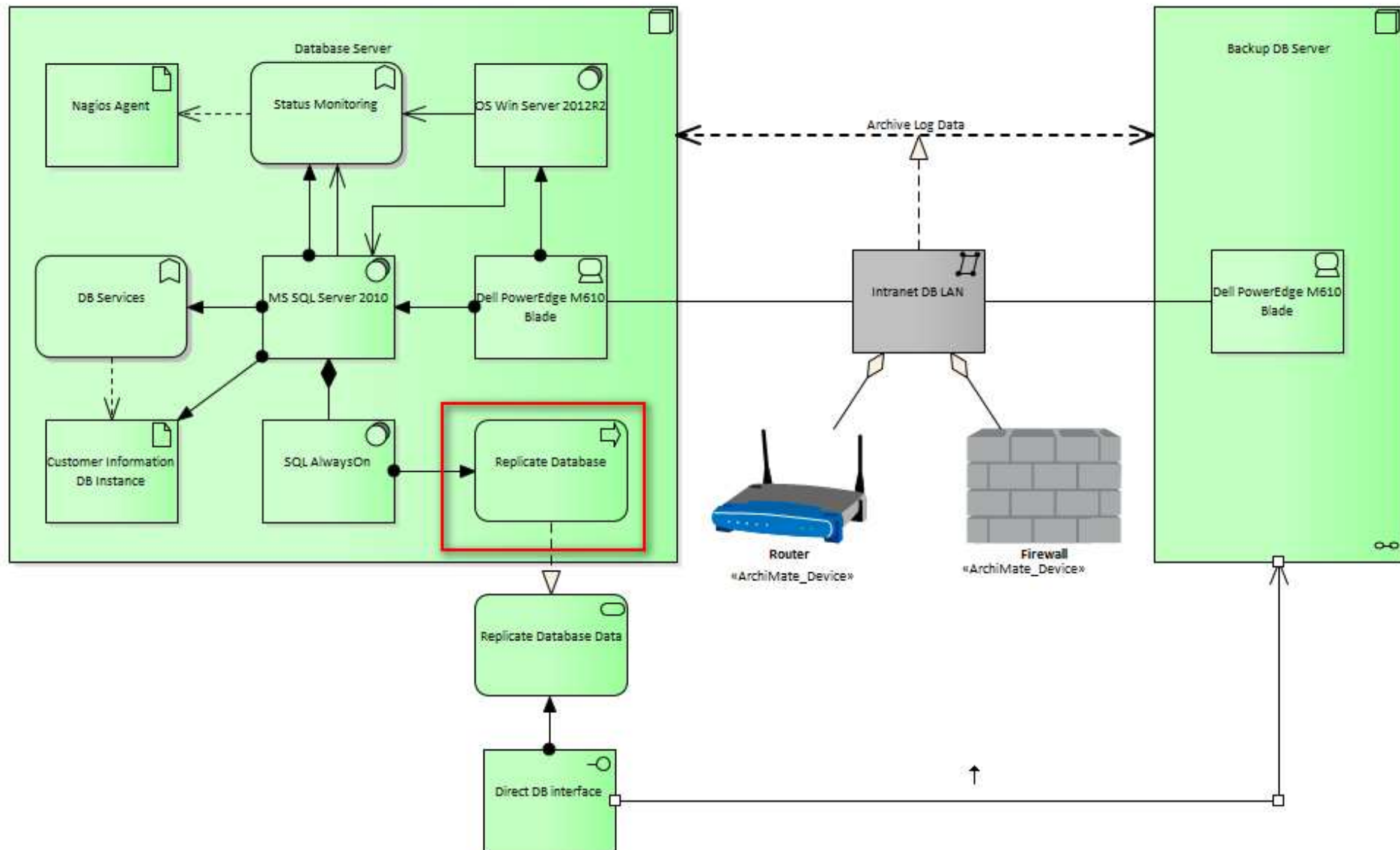
Definice:

Technologický proces (Technology Process) reprezentuje sekvenci akcí (či chování – Behavior) na technologické vrstvě (Technology Layer), které musí být splněny pro dosažení požadovaného výstupu.

Definice :

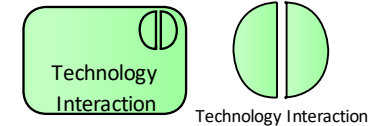
- Technology Process popisuje interní chování konkrétního Node (a jeho specializací – Device, Systém Software...), které je potřebné pro realizaci jedné či více technologických služeb (Technology Service).
- Technology Process je sekvence akcí vykonávaných Node v daném pořadí.
- Technology Process může spouštět a být spouštěn jinými Technology Process a dalšími technologickými behaviorálními elementy (Technology event, function, interaction) pomocí vazby Triggering.
- Technology Process může přistupovat k technologickým objektům a artefaktům (vytvářet, číst, měnit a mazat).
- Node (a jeho specializace – Device, Systém Software...) může být přiřazen (Assignment) k Technology Process, což znamená, že tento Node daný proces provádí.
- Technology Process je z pohledu okolního prostředí blackbox, proto internal behavior.
- Technology Process může realizovat jednu či více aplikačních služeb.
- Technology Process může být podporován (vazba Serving) aplikačními službami.
- Technology Process by měl obsahovat v názvu sloveso a popisovat účel procesu (Replicate Database).

Elementy v Technology Layer – Technology Process #2



Elementy v Technology Layer – Technology Interaction #1

Typ: Internal Behavior



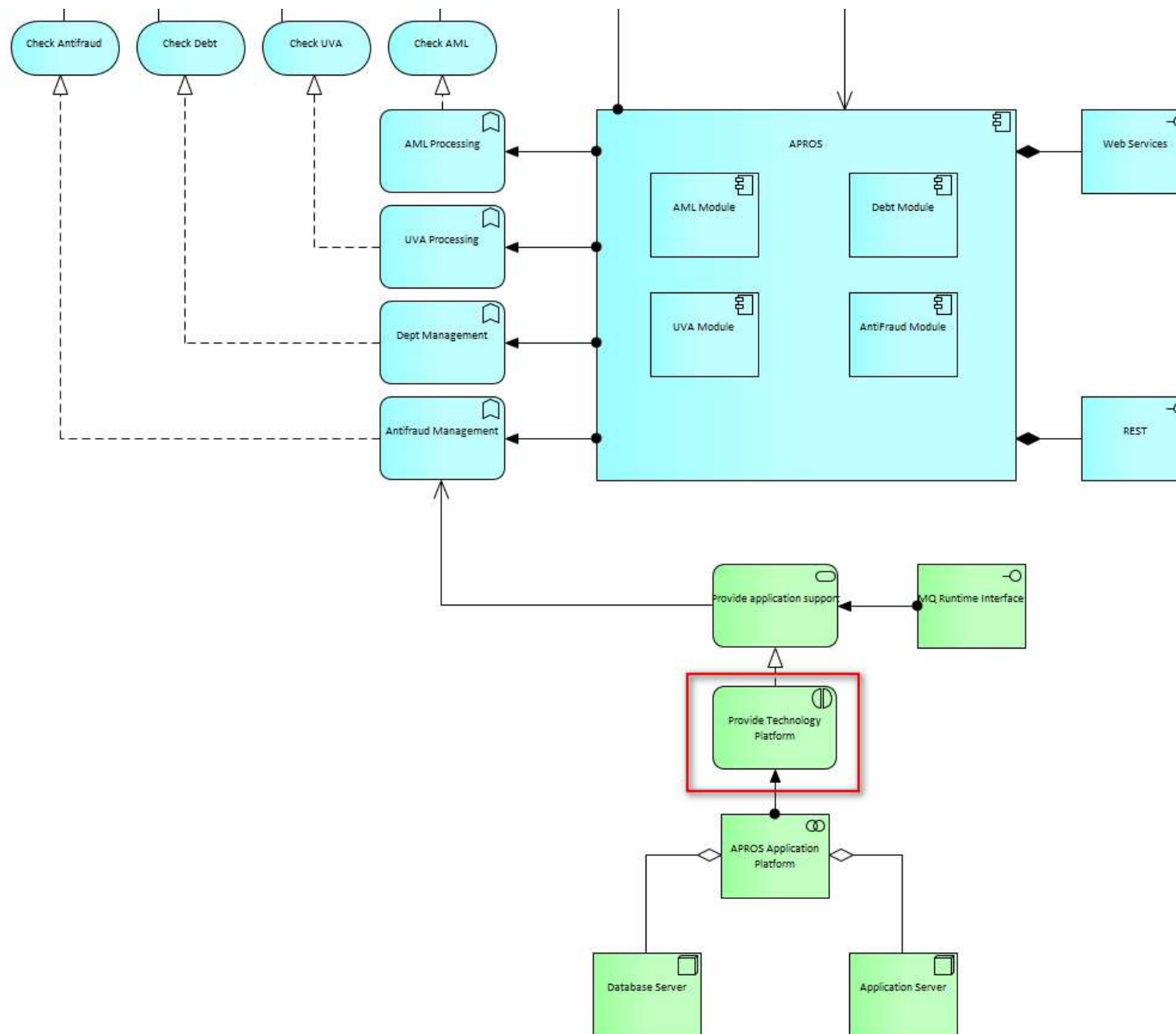
Definice:

Technologická interakce (Technology Interaction) reprezentuje konkrétní společné chování prováděné kolaborací (Technology Collaboration) jednoho či více uzlů (Node).

Definice :

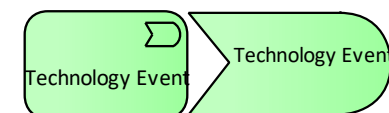
- Technology Interaction popisuje společné chování technologických uzlů (Node), sdružených v technologické kolaboraci (Technology Collaboration). Interakce popisuje toto chování minimálně do té míry detailu, která je potřeba pro realizaci technologické služby (Technology Service). Kolaborace je na technologickou interakci navázána přes vazbu Assignment.
- Pro větší úroveň detailu lze Technology Interaction popsat pomocí UML, například sekvenčními diagramy.
- Technology Interaction může realizovat technologickou službu (Technology Service).
- Technologické služby (Technology Service) mohou podporovat Technology Interaction (vazba Serving po směru šipky k Technology Interaction).
- Technology Interaction může přistupovat k technologickým objektům (Technology Object) a artefaktům.
- Název Technology Interaction by měl obsahovat sloveso a jasně vyjadřovat účel interakce, například Update Customer Data.
- Technology Interaction může vstupovat do všech standardních vztahů mezi technologickými behaviorálními elementy, podporovat je, být podporována a interagovat mezi nimi pomocí vazby Triggering.

Elementy v Technology Layer – Technology Interaction #2



Elementy v Technology Layer – Technology Event #1

Typ: Behavior

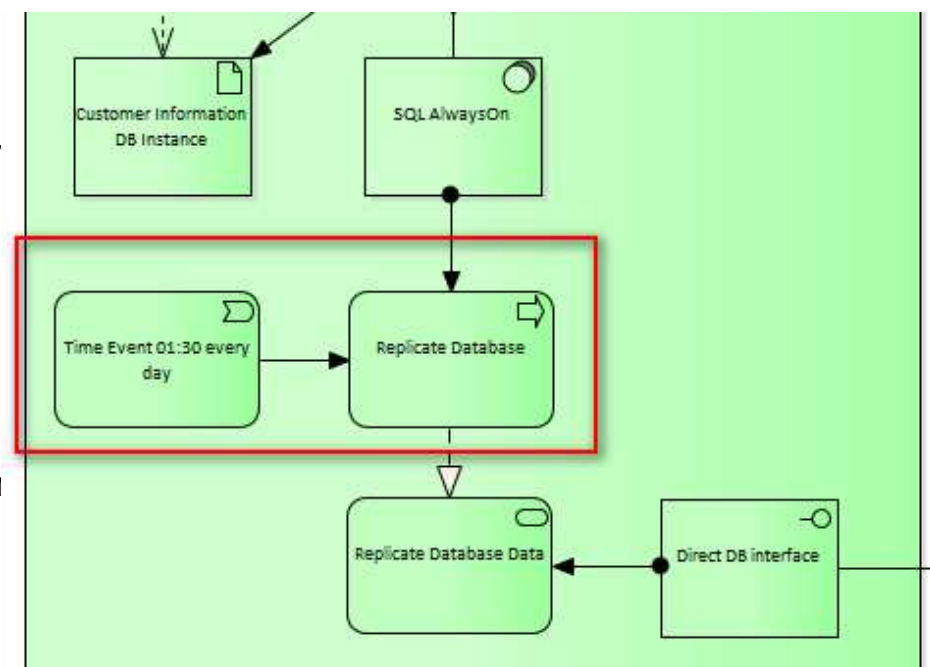


Definice:

Technology Event je aplikační chování, které představuje změnu stavu.

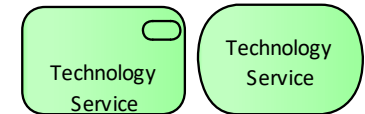
Definice :

- Na rozdíl od technologických procesů, funkcí a interakcí, Technology Event nemá trvání – je okamžitý (nicméně, může mít časový údaj, kdy je event spouštěn – přesný čas, datum nebo interval, den v týdnu či měsíci, opakování). Použitelné třeba pro automatizované replikace dat apod.
- Technology Process (a další technologické elementy chování – Technology Function, Interaction) mohou být spouštěny a přerušovány elementem Technology Event.
- Technology Event může nastat mimo organizaci (třeba externí událostí) nebo může být vyvolána technologickým procesem (funkcí, interakcí, ale třeba i jiným zařízením - Device).
- Technology Event může být složen z více Technology Event (událostí) pomocí vazby agregace nebo kompozice.
- V názvu by měla být činnost v dokonavém tvaru (data přijata).



Elementy v Technology Layer – Technology Service #1

Typ: External Behavior



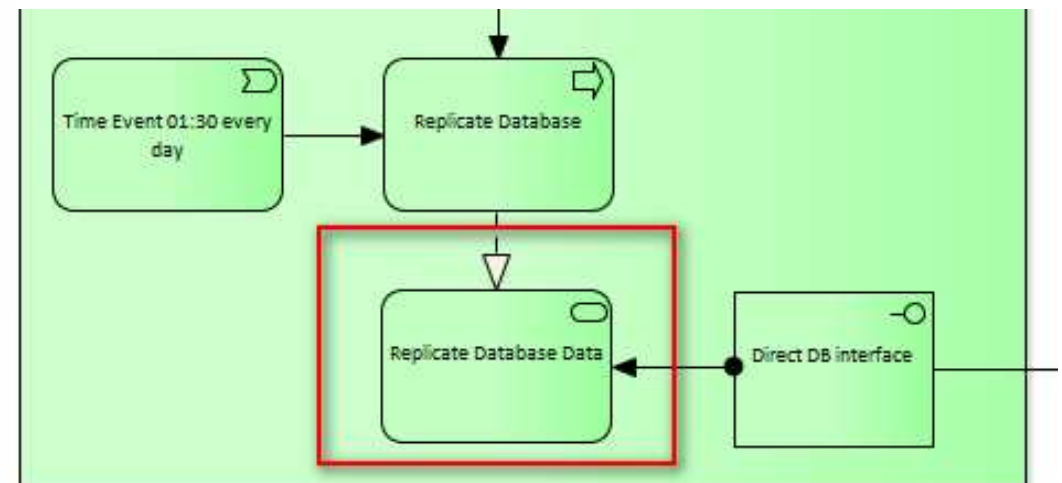
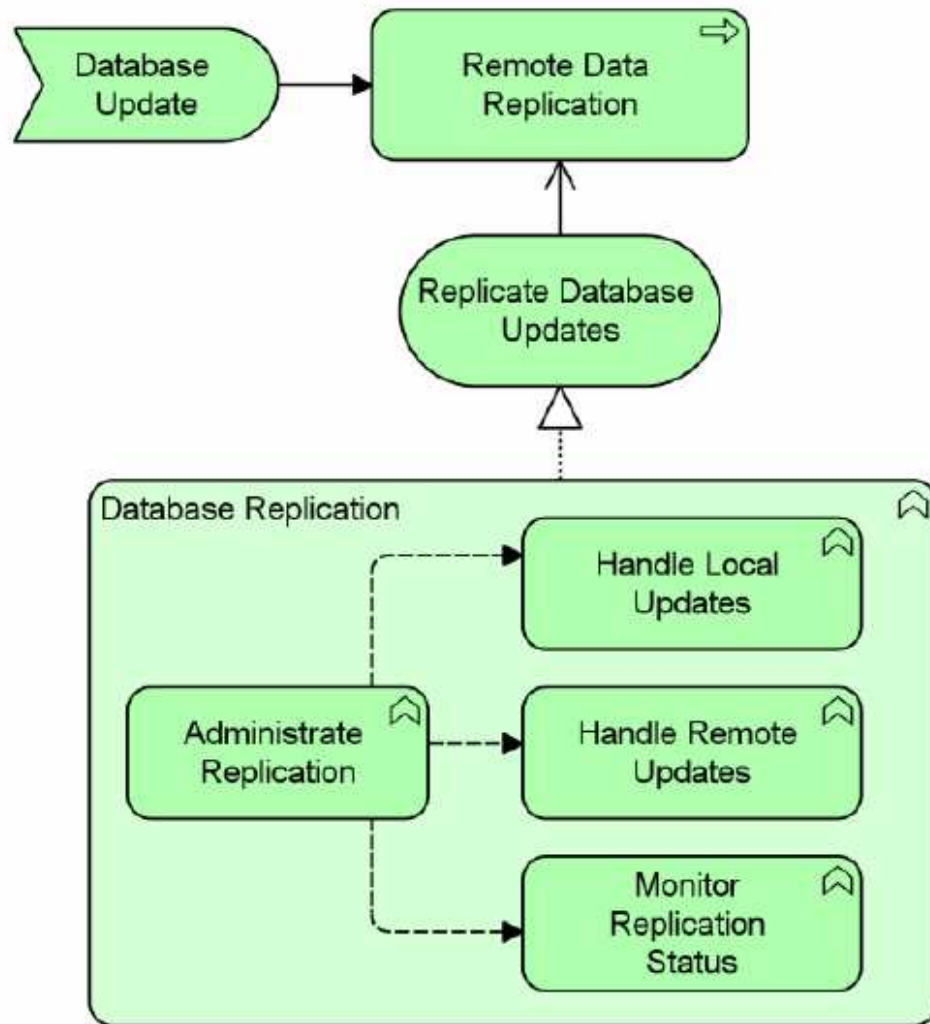
Definice:

Technologická služba (Technology Service) představuje explicitně definované chování (na technologické úrovni), které je vystaveno do infrastruktury.

Definice :

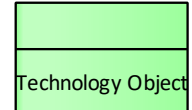
- Technology Service vystavuje funkcionalitu uzlů (Node) a jeho specializaci (Device, System software, Collaboration) do prostředí. Tato funkcionalita je dostupná prostřednictvím jednoho či více Technology Interface (vazbou Assignment) a jeden Technology Interface může zpřístupňovat více Technology Service.
- Technology Service je realizována jednou či více technologických funkcí či jiným technologickým behaviorálním elementem.
- Technology Service může vytvářet, číst, editovat a mazat technologické objekty (Technology Object) a artefakty.
- Technology Service by měla mít smysl z pohledu okolního prostředí a měla by poskytovat ucelenou a použitelnou službu svým uživatelům.
- Technology Service může podporovat (pomocí Serving směrem od Technology Service) technologické funkce, služby a behaviorální elementy z aplikační vrstvy (aplikační funkce, procesy, interakce, eventy).
- Název Technology Service by měl obsahovat sloveso a popisovat konkrétní činnost. V AJ se doporučuje použít přímo slovo service.

Elementy v Technology Layer – Technology Service #2



Elementy v Technology. Layer – Technology Object #1

Typ: Passive Structure



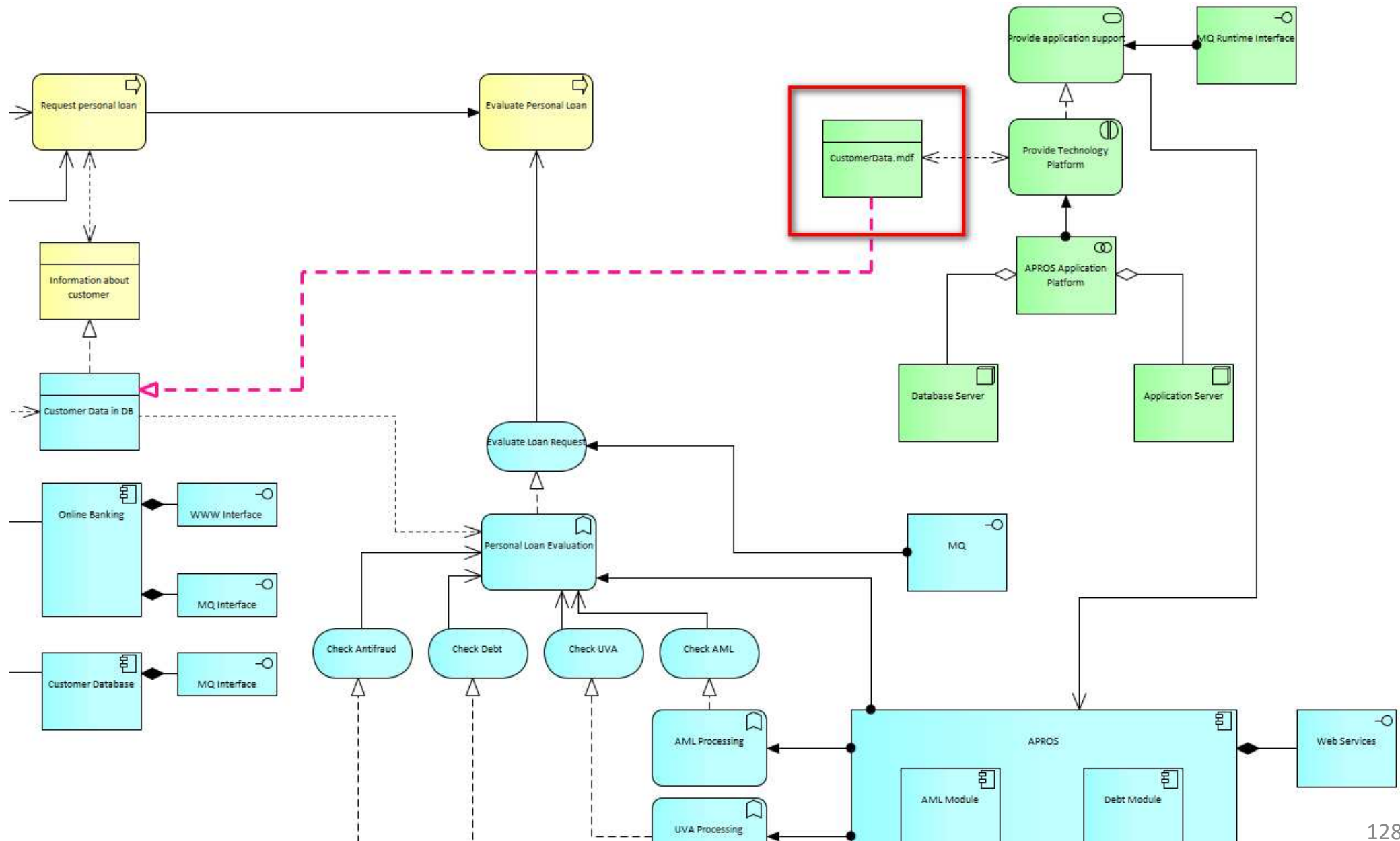
Definice:

Technology Object reprezentuje pasivní element, který je vytvářen či zpracováván behaviorálními technologickými elementy.

Definice :

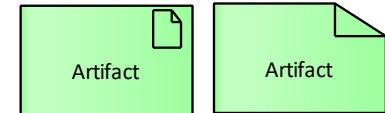
- Technology Object reprezentuje fyzické objekty (datové soubory apod.), se kterými se manipuluje v rámci infrastruktury společnosti..
- Technology Object jsou pasivní, nemohou tedy spouštět (Trigger) technologické procesy, funkce apod.
- Technology Object odpovídají nejlépe třídám v jazyce UML (bez operací a atributů), pouze ve výjimečných případech jsou obdobou instancí.
- K Technology Object mohou přistupovat technologické behaviorální elementy – Technology Function, Process, Interaction, Service, Event. Přístupem se rozumí jejich vytváření, čtení, editace a rušení.
- Technology Object mohou být skládány z jiných Technology Object pomocí agregací nebo kompozic, mohou mezi sebou být navázány pomocí asociací nebo mohou být specializovány (pomocí vazby specializace – Specialization.)
- Technology Object mohou realizovat datové objekty (Application Data Object) na aplikační vrstvě (a elementy z nich specializované).
- Technology Object může být realizován artefaktem (Artifact) nebo materiálem (Material).
- Jméno Technology Object by mělo obsahovat podstatné jméno.

Elementy v Technology. Layer – Technology Object #2



Elementy v Technology. Layer – Technology Artifact #1

Typ: Passive Structure



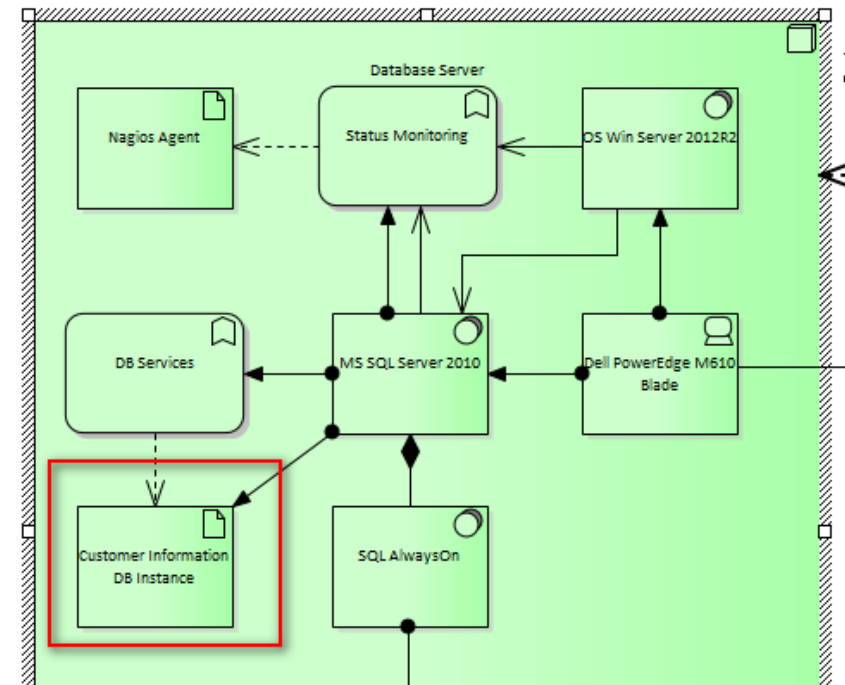
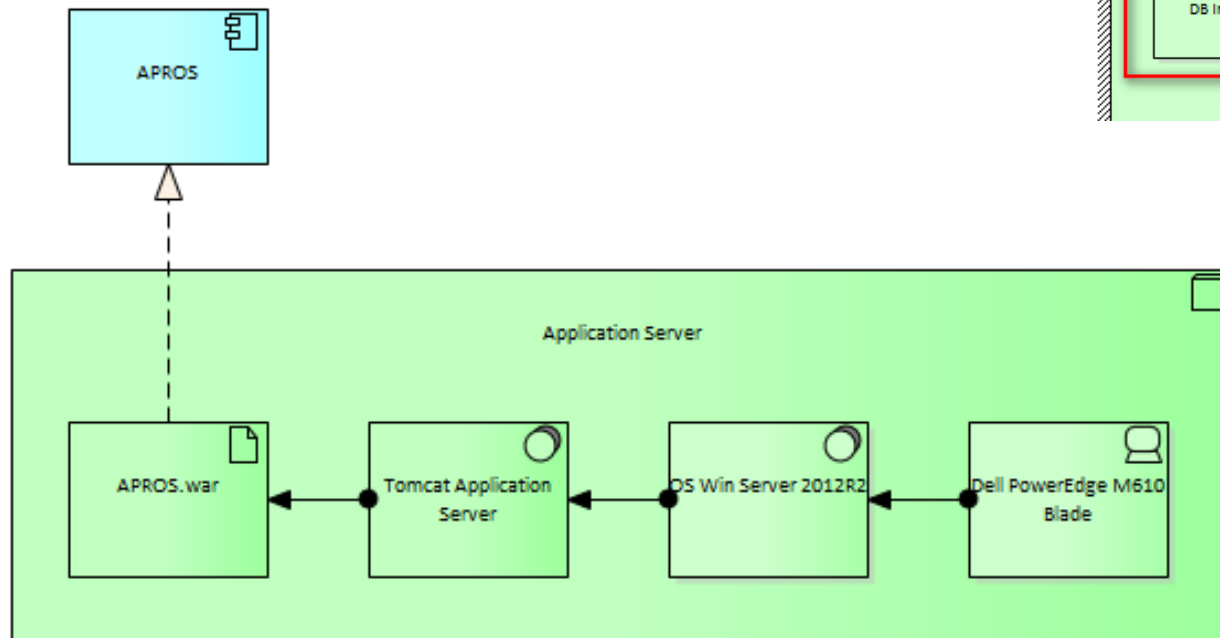
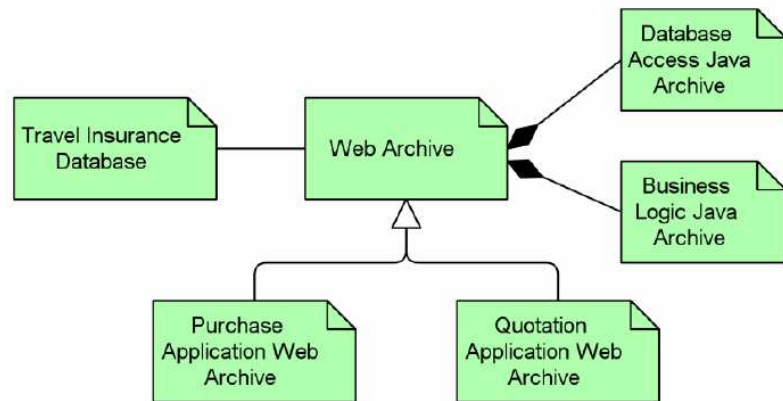
Definition:

Artifact reprezentuje množinu (či kousek) dat, které jsou využívána nebo produkována v procesu vývoje software či nasazením (deployment) a činností IT systému.

Definition :

- Artifact reprezentuje hmatatelný element v IT světě (fyzický soubor například).
- Artifact je specializací Technology Object.
- Artifact se typicky používá pro modelování konkrétních výstupů, jako jsou zdrojové kódy, spustitelné soubory a scripty, databázové tabulky, zprávy, dokumenty, specifikace nebo třeba soubory s modely.
- **Instance** artefaktu (Artifact) může být nasazena (deploy) na Node – pozor, nasazuje se instance, nikoliv třída).
- Artifact může být použit jako fyzická reprezentace datového objektu z aplikační vrstvy a tedy Data Object může být realizován jedním či více artefaktem (Artifact)..
- Aplikační komponenta může být realizována jedním či více artefaktem (Artifact).
- Systémový software (Technology System Software) může být realizován jedním či více artefaktem (Artifact).
- Node může být přiřazen (vazbou Assignment) k artefaktu pro znázornění, že je tento na něm nasazen (deployed).
- Typicky se Artifact používá pro modelování spustitelného kódu nebo dat. *Doporučení Open Group: O tyto elementy specializované z Artifactu, by mohla být rozšířena struktura ArchiMate.* (pozn. Proč to neudělají přímo? Možná v další verzi).
- Artifact mohou být skládány z jiných Artifact pomocí agregací nebo kompozic, mohou mezi sebou být navázány pomocí asociací nebo mohou být specializovány (pomocí vazby specializace – Specialization).
- Název artefaktu (Artifact) by měl být název souboru, který reprezentuje (například EA.exe).

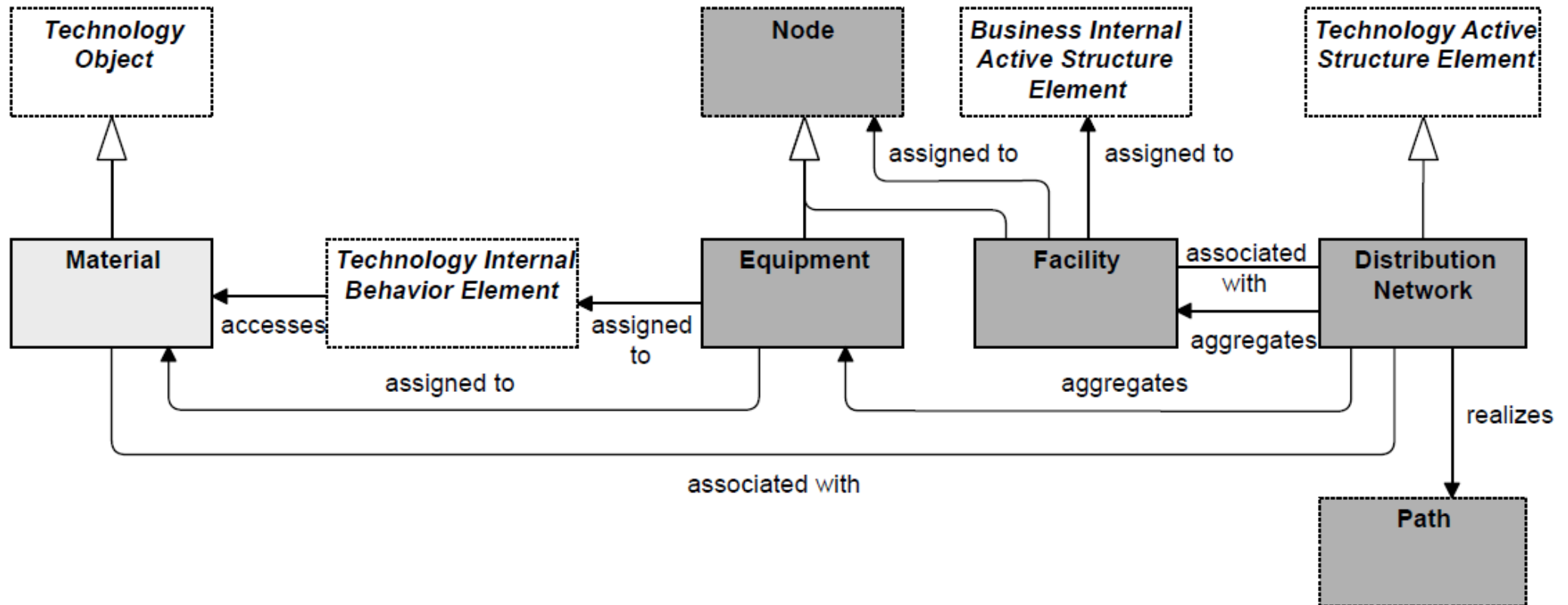
Elementy v Technology. Layer – Technology Artifact #2



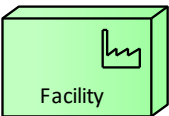
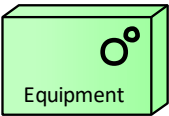
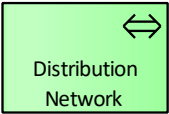
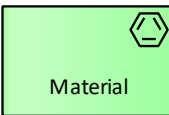
ArchiMate

Elementy – Physical Extension

Physical Extension- metamodel

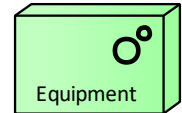


Elementy v Physical Extension – přehled #1

Element	Definice	Aspekt
 Facility	A facility represents a physical structure or environment.	Active structure
 Equipment	Equipment represents one or more physical machines, tools, or instruments that can create, use, store, move, or transform materials.	Active structure
 Distribution Network	A distribution network represents a physical network used to transport materials or energy.	Active structure
 Material	Material represents tangible physical matter or physical elements.	Passive structure

Elementy v Physical Extension – Equipment #1

Typ: Active Structure

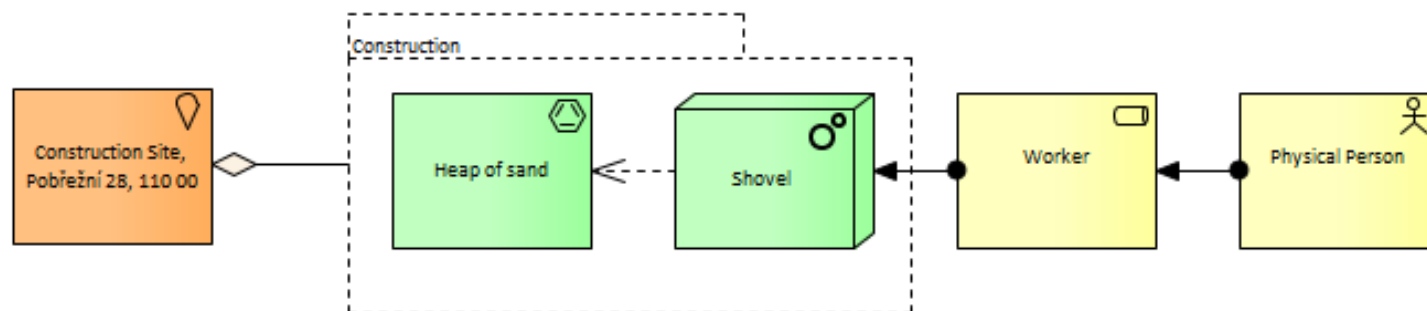


Definice:

Equipment reprezentuje jeden či více fyzický nástroj, stroj či zařízení, které může vytvářet, používat, ukládat, přesouvat nebo měnit materiál (Material).

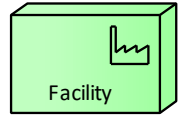
Definice :

- Equipment dokáže manipulovat s předměty ve fyzické realitě.
- Equipment je specializací Node z technologické vrstvy. Je tedy možné modelovat Node, které kombinují IT infrastrukturu (Device, System Software) a fyzickou infrastrukturu (Equipment), například CT scanner či rentgen v nemocnici, elektrárnu s ovládacími systémy a podobně.
- Equipment může přistupovat (vytvářet, měnit, používat, ukládat, přesouvat) Material (který je specializací Technology Object).
- Equipment může podporovat (pomocí vazby Serving) jiný Equipment a rovněž jakékoliv aktivní struktury z jakékoliv vrstvy, tedy například Business Actor, Role, Application Component, Node).
- Equipment může být přiřazen (Assignment – šipkou k Equipment) kterémukoliv aktivnímu elementu z jakékoliv vrstvy a naopak, například System Software může být (šipkou k System Software) přiřazen k Equipment pro znázornění, že Equipment může mít nějaký operační systém apod.
- Equipment může být složen z jiných Equipment pomocí Aggregation nebo Composition.
- Equipment může být přiřazen (šipkou k Equipment) k Facility pro znázornění, že je tam instalován nebo používán.
- Equipment může být agregován do Location.
- Equipment by měl mít v názvu podstatné jméno (Hammer, Car, Crane).



Elementy v Physical Extension – Facility #1

Typ: Active Structure



Definice:

Facility reprezentuje fyzickou strukturu nebo prostředí.

Definice :

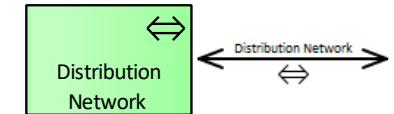
- Facility je specializací Node z technologické vrstvy. Reprezentuje fyzický zdroj, který je schopen využívat Equipment (skladovat, používat, namontovat na apod.).
- Typicky se používají na modelování továren, výrobních hal, skladů, budov či venkovních konstrukcí (které hrají nějakou důležitou roli pro procesy organizace – například pro výrobu, nebo distribuci).
- Příklady: továrna, laboratoř, sklad, prodejna, jeskyně nebo kosmická loď.
- Facility mohou být navzájem propojeny pomocí vazby (nebo elementu) Distribution Network.
- Facility může být složen z jiných Facility pomocí Aggregation nebo Composition.
- Equipment může být přiřazen (šipkou k Equipment) k Facility pro znázornění, že je tam instalován nebo používán.
- Facility může podporovat (pomocí vazby Serving) jiný Facility a rovněž jakékoliv aktivní struktury z jakékoliv vrstvy, tedy například Business Actor, Role, Application Component, Node).
- Facility může být agregován do Location.
- Facility by měl mít v názvu podstatné jméno odkazující se na typ Facility (Ostravsko – Karvinský uhelný důl Vítěz).

Elementy v Physical Extension – Distribution Network #1

Typ: Active Structure / Realizace vazby Path

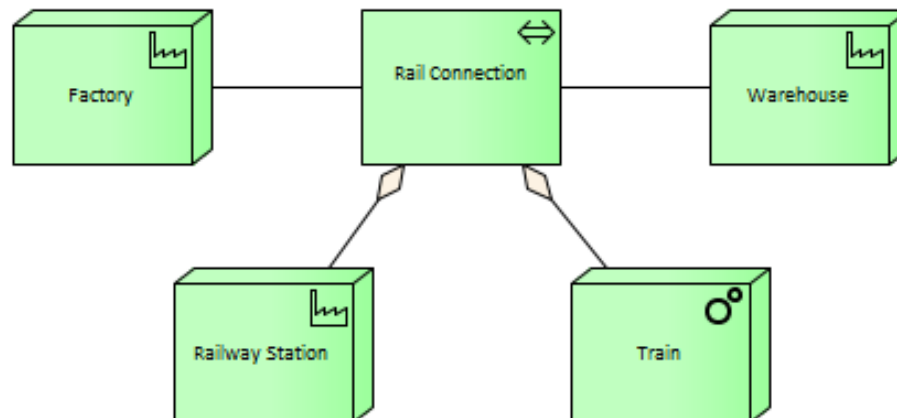
Definice:

Distribution Network reprezentuje fyzickou dopravní síť, pomocí níž lze přepravovat materiál (Material) nebo energii.



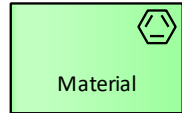
Definice :

- Distribution Network je realizací logických cest (Path) mezi Node.
- Distribution Network může být složen z jiných Distribution Network (subnetworks) pomocí Aggregation nebo Composition.
- Distribution Network propojuje dva, nebo více Node (a jejich specializací, například Facility).
- Distribution Network může realizovat jedno nebo více logických propojení (Path).
- Distribution Network může agregovat Facility a Equipment, například, Nádraží a vlak mohou být částmi Distribution Network s názvem Železniční spojení. Je to obdoba Communication Network z Technology Layer.
- Distribution Network by měl obsahovat podstatné jméno.



Elementy v Physical Extension – Material #1

Typ: Passive Structure

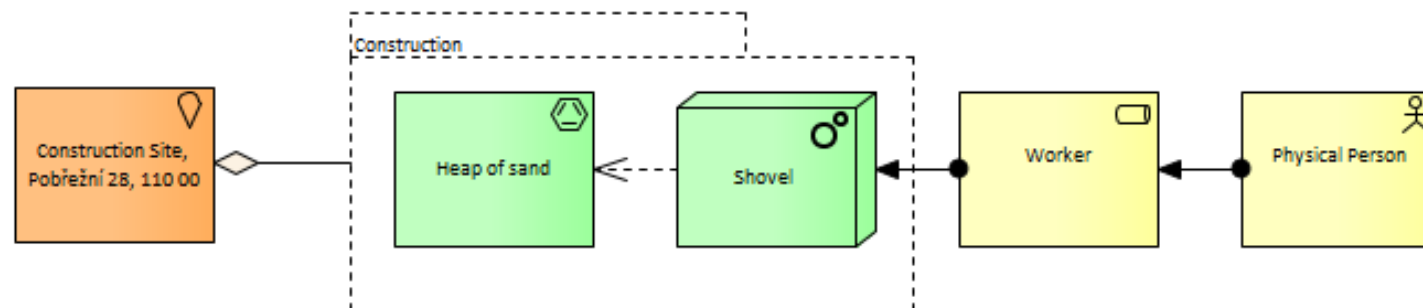


Definice:

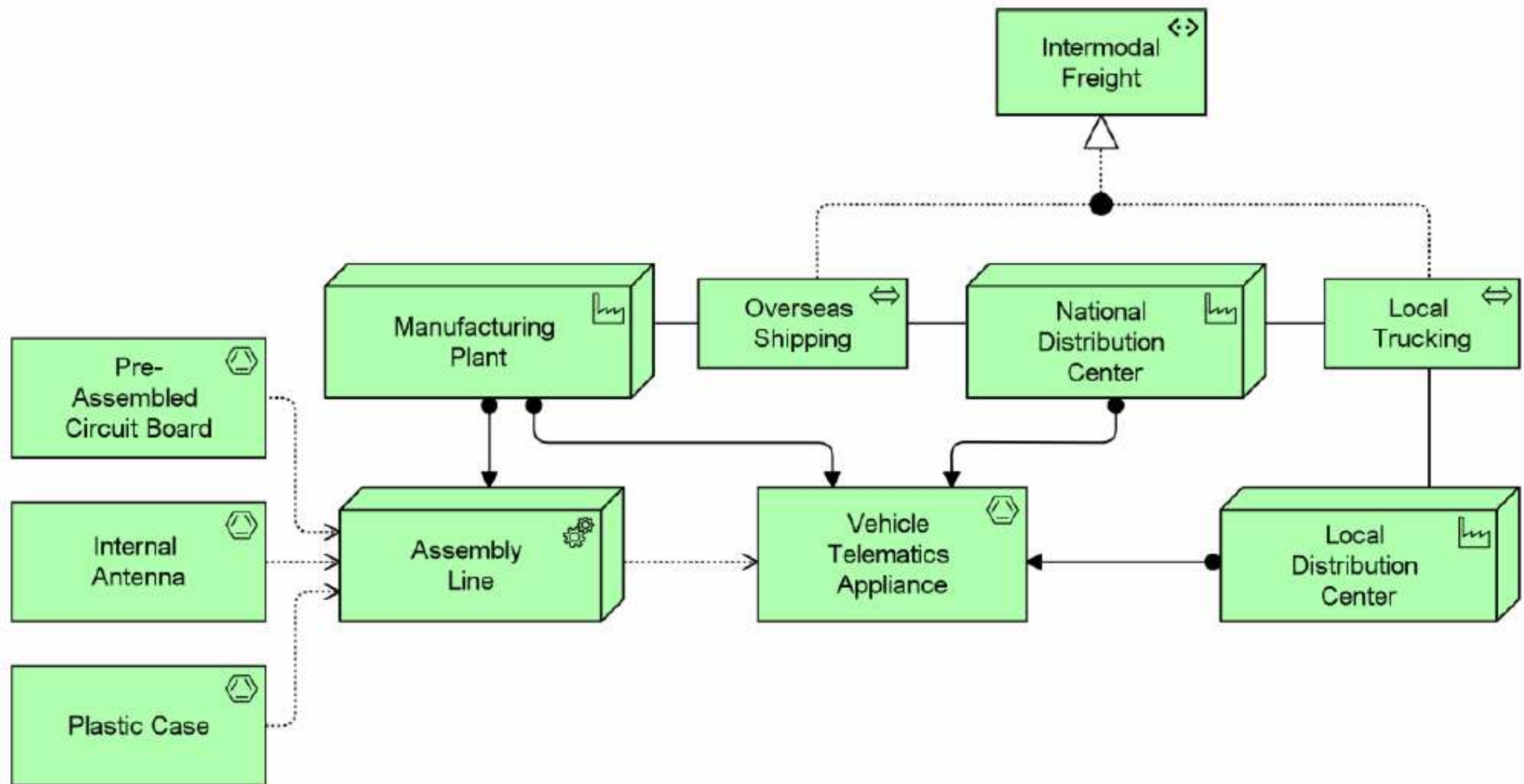
Material reprezentuje hmatatelný fyzický element.

Definice :

- Material reprezentuje hmatatelnou podstatu elementů s takovými atributy, jako je třeba váha, rozměry apod.
- Material je typicky používán pro modelování materiálů, fyzických produktů nebo třeba energetických zdrojů, jako je třeba palivo.
- Material může být složen z jiných Material pomocí Aggregation nebo Composition.
- Material by měl obsahovat podstatné jméno.



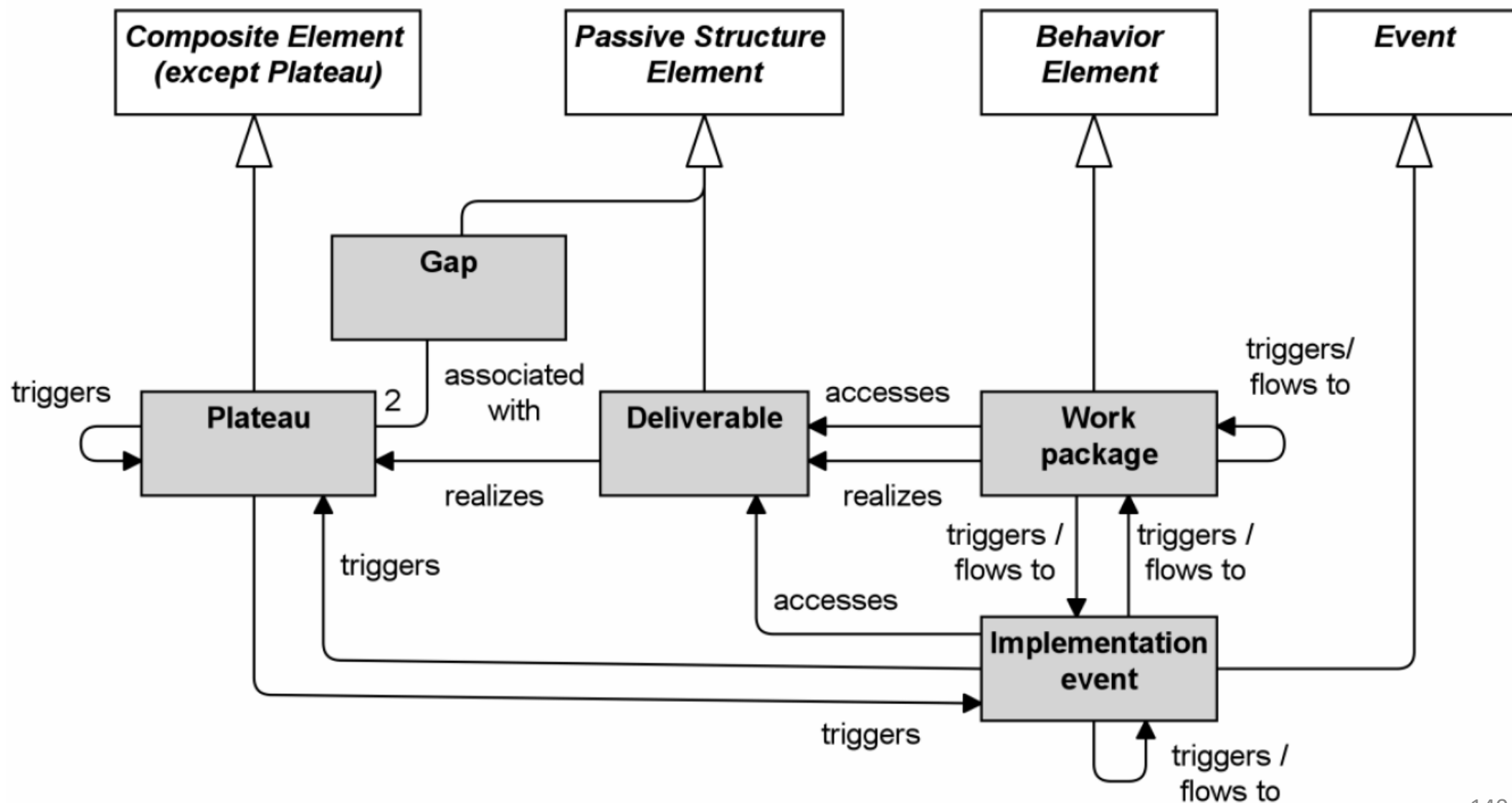
Elementy v Physical Extension – Example



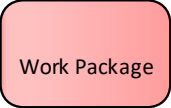
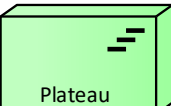
ArchiMate

Elementy – Implementation & Migration Extension

Implementation & Migration Ext. - metamodel



Elementy v Impl. & Migr. Extension – přehled #1

Element	Definice	Aspekt
 Work Package	A work package represents a series of actions identified and designed to achieve specific results within specified time and resource constraints.	Behavior
 Implementation Event	An implementation event is a behavior element that denotes a state change related to implementation or migration.	Behavior
 Gap	A gap represents a statement of difference between two plateaus.	Passive structure
 Deliverable	A deliverable represents a precisely-defined outcome of a work package.	Passive structure
 Plateau	A plateau represents a relatively stable state of the architecture that exists during a limited period of time.	Passive Composite

Elementy v Implementation & Migration Extension – Work Package #1

Typ: Behavior

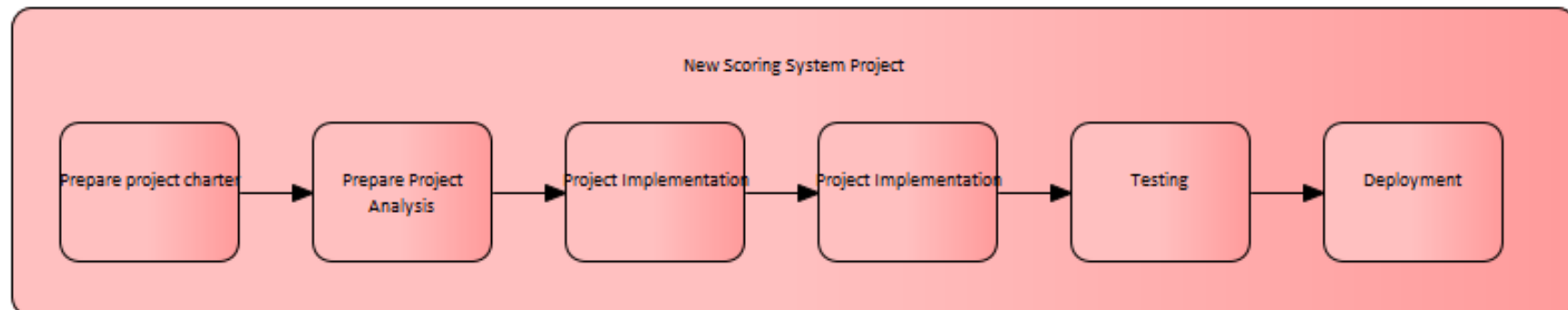
Work Package

Definice:

Work Package reprezentuje posloupnost akcí, která je potřebná pro dosažení definovaných výsledků během definovaného času a za respektování omezení zdrojů (Resource Constraints).

Definice :

- Work Package je klíčový element chování (Behavior) této extenze. Tento element definuje klíčové parametry, jako jsou začátek a konce práce (implementace) a realizuje definovaný soubor (set) cílů (Goal) a dodávek (Deliverable) či výstupů.
- Work Package může být využit pro modelování dílčích projektů (sub-projects) nebo úkolů uvnitř projektu, kompletních projektů, programů nebo projektových portfolií).
- Konceptně se Work Package podobá business procesům. Může obsahovat sadu definovaných úkolů, zaměřených na dosažení definovaného výsledku. Nicméně, na rozdíl od procesů, které jsou principiálně opakovatelné (nejedná se o instance) tak Work Package je unikátní a jednorázový (tedy konkrétní projekt apod.). Nicméně, stále může být Work Package definován způsobem - velmi podobným definici procesu.

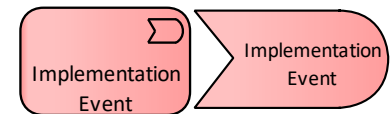


Elementy v Implementation & Migration Extension – Implementation Event #1

Typ: Behavior

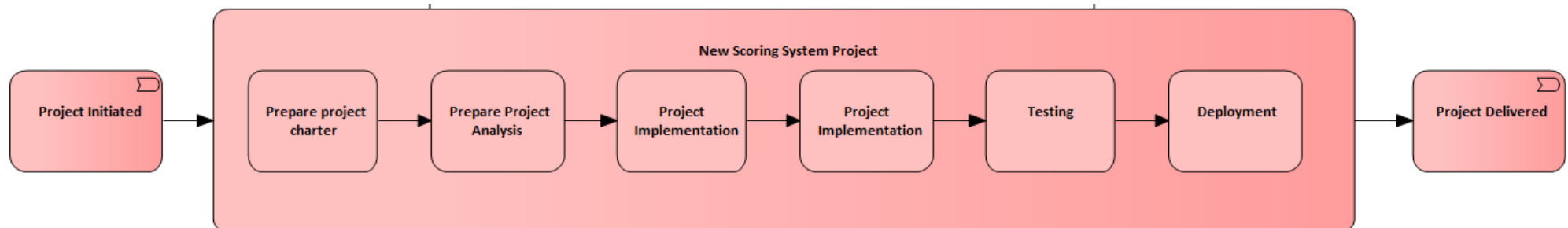
Definice:

Implementation Event představuje element chování (Behavior), který popisuje změnu stavu, vztaženou k implementaci a migraci.



Definice :

- Work Package může být spouštěn nebo přerušován událostmi Implementation Event. Work Package také může spouštět události (Implementation Event), které poté spouští jiné chování (Behavior).
- Implementation Event nemá, na rozdíl od Work Package dobu trvání, je okamžitý (respektive, v okamžiku, kdy Event nastane, událost již proběhla).
- Implementation Event může mít časový údaj, kdy je event spouštěn – přesný čas, datum nebo interval, den v týdnu či měsíci, opakování). To může být užitečné při modelování projektového plánu, milníků.
- Implementation Event může být vyvolán i dokončením příslušného Work Package (pomocí vazby Triggering).
- Implementation Event může popisovat například změnu v životním cyklu projektu nebo jeho výstupu (Deliverable).
- Implementation Event může přistupovat (vytvářet, číst, editovat a mazat) Deliverable. Tedy například, Implementation Event „Release to production“ může přistupovat (vazba Access) k Deliverable „Final Build“, „Staging Environment“ a „Production Environment“.
- Implementation Event může být složen z jiných Implementation Event (pomocí vazeb Aggregation a Composition).
- Implementation Event může být spouštěn (Triggered) z Work Package, jiné Implementation Event nebo z Plateau.
- Implementation Event může být asociován (Association) k jakémukoliv Core elementu (například aplikační komponentě), aby bylo možno zaznamenat změnu jejího stavu (Lifecycle).
- Implementation Event by mělo obsahovat dokonavé sloveso a co nejlépe popisovat stav: Projekt byl iniciován, systém byl dodán, systém byl vyřazen, systém byl nasazen do produkčního prostředí.



Elementy v Implementation & Migration Extension – Deliverable #1

Typ: Passive Structure

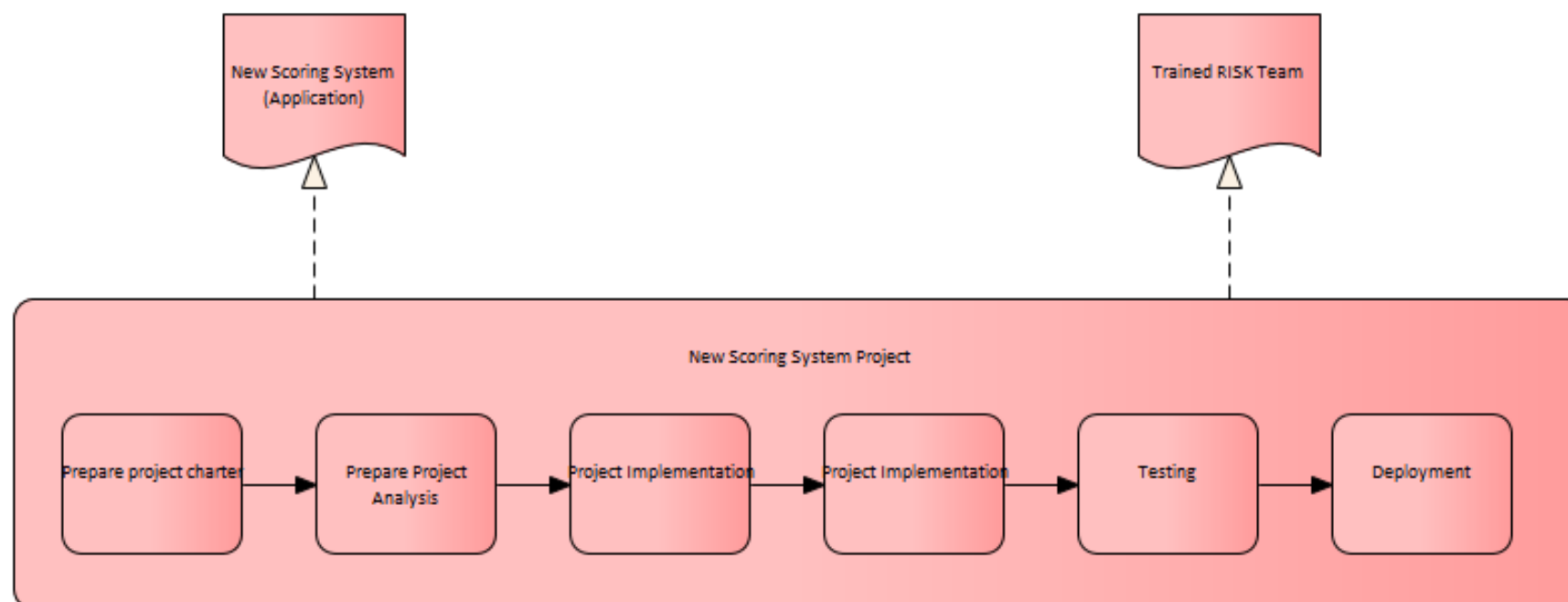
Deliverable

Definice:

Deliverable reprezentuje přesně definovaný výstup Work Package.

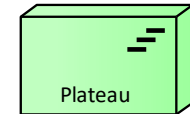
Definice :

- Work Package produkuje Deliverable.
- Deliverable může být jakýkoliv výstup – reporty, dokumenty, služby, software, fyzický produkt, organizační změna společnosti nebo část implementace informační architektury.
- Deliverable často bývá předmětem kontraktu nebo bývá formálně revidována, potvrzována a podepisována stakeholdery (tak jak to, například, definuje TOGAF).
- Element Deliverable může být realizován (vazba Realization) elementem Work Package.



Elementy v Implementation & Migration Extension – Plateau #1

Typ: Composite

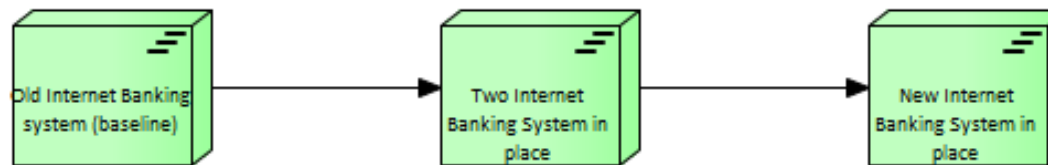


Definice:

Plateau reprezentuje relativně stabilní stav informační architektury (či její části), která existuje po vymezený časový úsek.

Definice :

- Jeden z důležitých principů TOGAFu je to, že různé architektury (business, datová, aplikační) jsou popsány v různých časových okamžicích. V každé fázi B, C a D ADM cyklu je například zakreslen výchozí stav příslušné architektury a rovněž cílový stav. Ve fázi E ADM cyklu (Opportunities and Solutions) jsou definovány takzvané průběžné (Transition) architektury, které popisují inkrementální změny mezi výchozí a cílovou architekturou.
- Průběžné architektury jsou důležité, aby ukázaly přínos jednotlivých projektů (Work Package) a jejich přidanou hodnotu pro business.
- Pro podporu tohoto principu je definován právě element Plateau.
- Plateau představuje právě jeden stav architektury (výchozí, průběžné nebo cílové) v čase.
- Do Plateau se mohou agregovat Core elementy pro znázornění, že patří do dané architektury.
- Název Plateau by měl obsahovat podstatné jméno a popisovat stav architektury.



Elementy v Implementation & Migration Extension – GAP #1

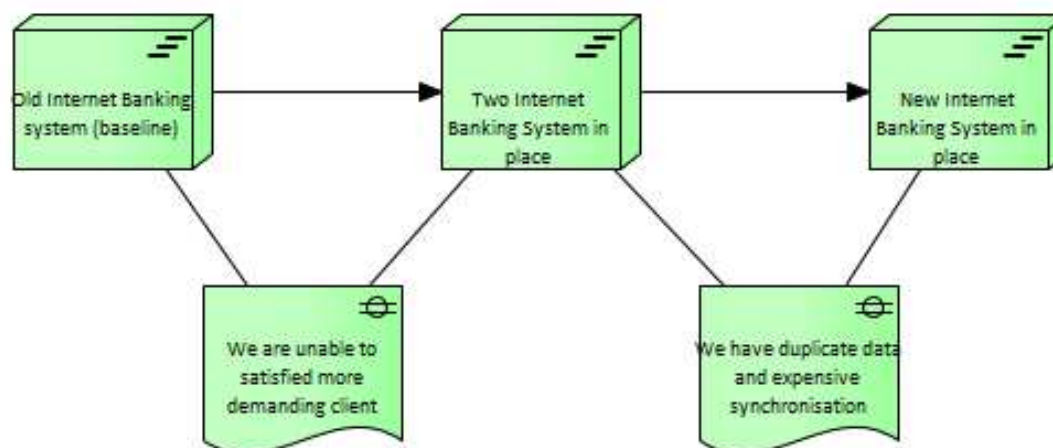
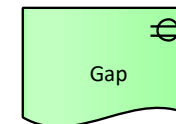
Typ: Passive Structure

Definice:

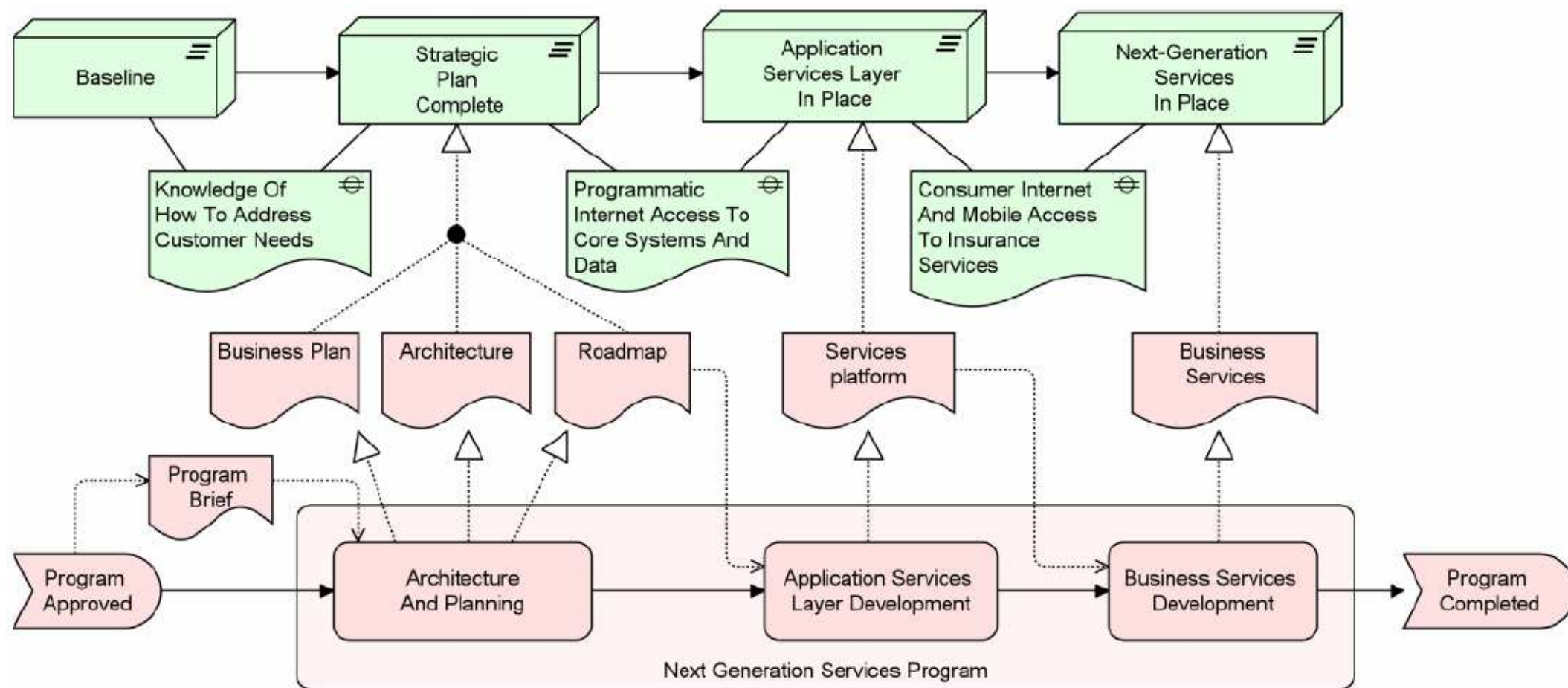
Gap reprezentuje rozdíl mezi dvěma stavy architektury (Plateau).

Definice :

- Gap element je asociován (Association) se dvěma architekturami a reprezentuje rozdíl mezi nimi.
- Podle frameworku TOGAF je Gap velmi důležitý výstup fází B, C a D fází ADM procesu a slouží jako jeden ze vstupů pro implementaci a migraci.



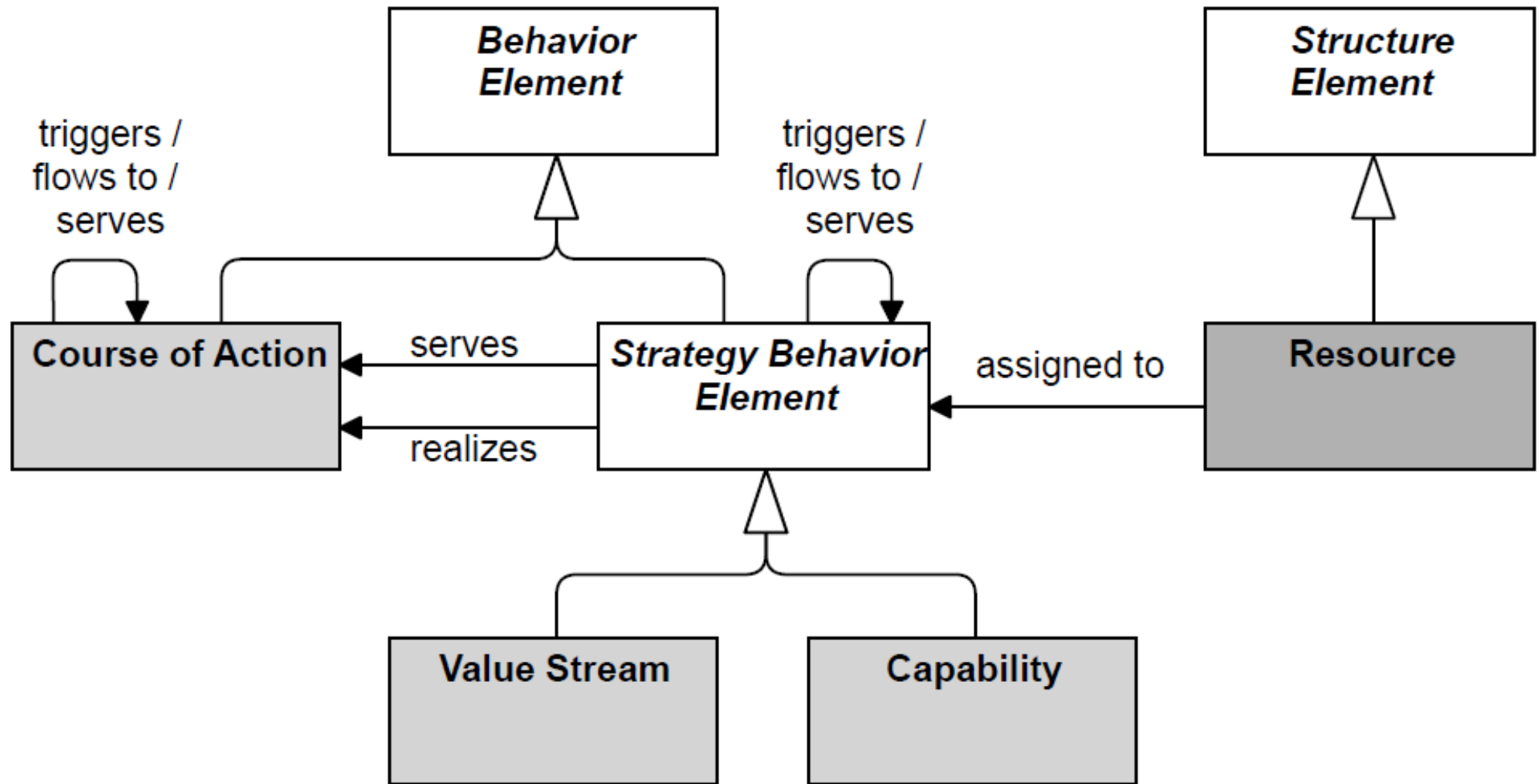
Elementy v Implementation & Migration Extension – Example #1



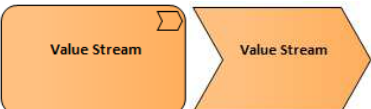
ArchiMate

Elementy – Strategy Extension

Strategy Extension - metamodel



Elementy ve Strategy Extension – přehled #1

Element	Definice	Aspekt
	A course of action is an approach or plan for configuring some capabilities and resources of the enterprise, undertaken to achieve a goal.	Behavior
	A capability represents an ability that an active structure element, such as an organization, person, or system, possesses.	Behavior
	A resource represents an asset owned or controlled by an individual or organization.	Passive structure
	A value stream represents a sequence of activities that create an overall result for a customer, stakeholder, or end user.	Behavior

Elementy v Strategy Extension – Course Of Action #1

Typ: Behavior



Definice:

Course Of Action je reprezentace konkrétního přístupu či plánu pro konfiguraci (či využití) některých Business Capability a Resource (zdrojů) organizace (či fyzické osoby), pro dosažení stanoveného cíle.

Definice :

- Course Of Action představuje rozhodnutí organizace (nebo fyzické osoby – souhrnně Enterprise), co se bude dělat.
- Course Of Action lze kategorizovat na strategie a taktiky. Jakkoliv nelze přesně rozlišit mezi těmito dvěma, dá se říci, že strategie bývají dlouhodobé a se širokým záběrem, kdežto taktiky spíše krátkodobé a se zúženým záběrem.

Elementy v Strategy Extension – Value Stream #1

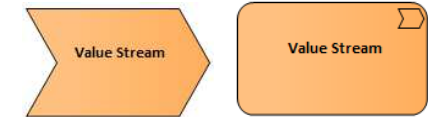
Typ: Behavior

Definice:

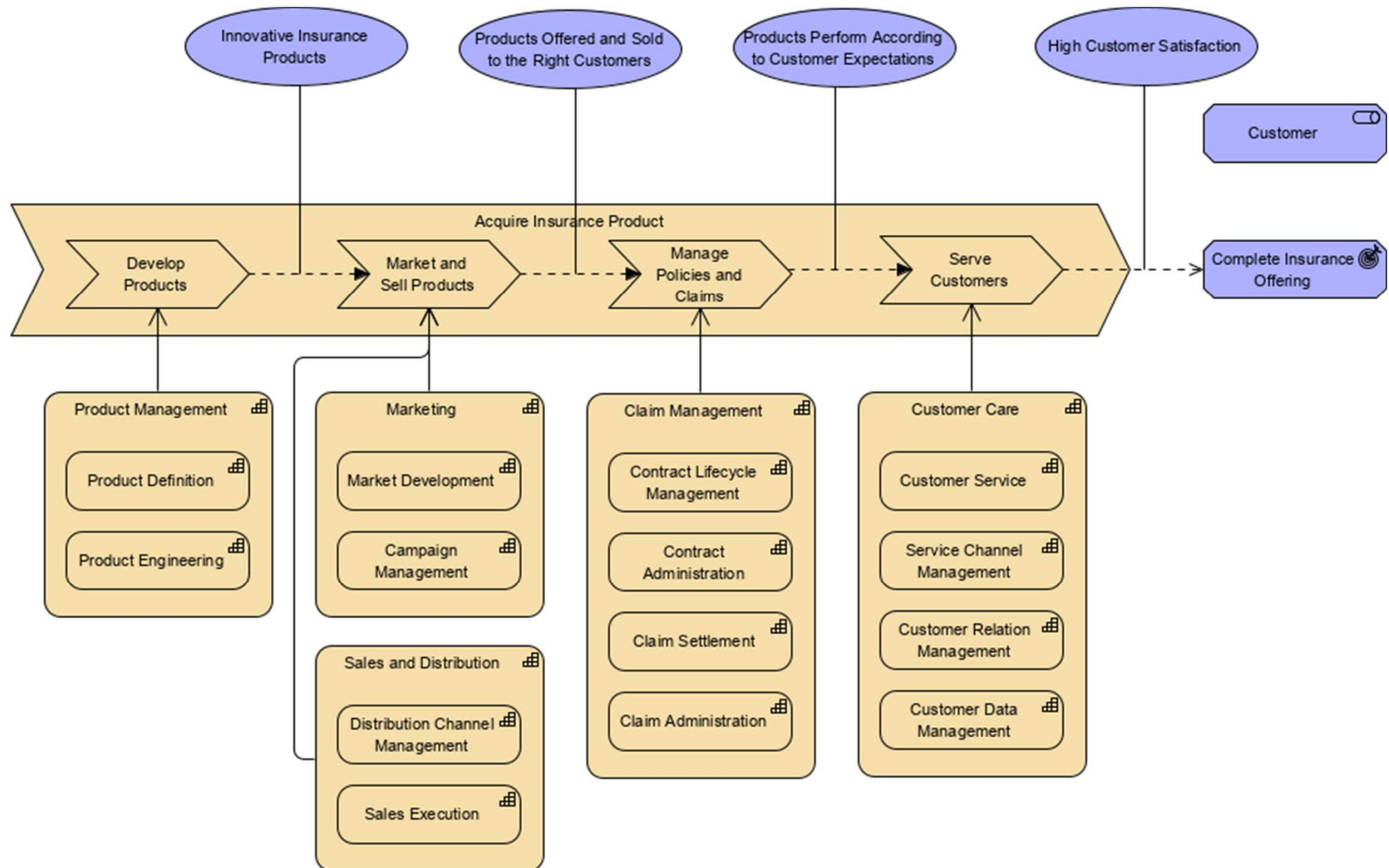
Value Stream reprezentuje sekvenci aktivit, které je potřeba vykonat, aby bylo dosaženo požadovaného výsledku či výstupu (směrem k zákazníkovi, stakeholderovi nebo konečnému uživateli).

Definice :

- Value stream je typicky realizován business procesem, ale je možné použít i jiné core elementy.
- Jeden Value stream může být realizován více elementy (například procesy) a jeden proces (či jiný element) může realizovat více value stream elementů.
- Value stream může být definován na úrovni celé společnosti, nebo na úrovni oddělení, týmů apod. a to směrem k interním či externím zákazníkům.
- Value stream se modeluje z pohledu cílového příjemce – stakeholdera, zákazníka apod.
- Value stream, oproti business procesům, se modeluje na vyšší úrovni a dívá se na společnost z pohledu hodnot – tedy, jaké aktivity musí být vykonány, nikoliv jak jsou vykonávány. Neobsahuje žádné rozhodování či alternativní toky ale popisují důležitost aktivity a její hodnotu.
- Název Value Stream by měl obsahovat sloveso.



Elementy v Strategy Extension – Value Stream #2



Elementy v Strategy Extension – Business Capability #1

Typ: Behavior



Definice:

Business Capability (schopnost, kompetence) reprezentuje nějakou schopnost, kterou má Active Structure Element, jako je třeba organizace, osoba nebo systém.

Definice :

- V businessu se často stává, že výstupy (strategie a high-level cíle) strategického plánování nejsou přímo implementovatelné v architektuře společnosti. Tyto dlouhodobé a obecné plány musí být blíže specifikovány až na úroveň, kdy z nich vyplývají proveditelné akce a to v podobě, že s nimi mohou pracovat jak business manažeři tak Enterprise Architect.
- Business Capability mohou pomoci tento problém, s obecností strategických plánů, vyřešit a to tak, že se zaměřují na business výstupy. Na jednu stranu poskytují pohled na aktuální a také na cílové schopnosti (capability) organizace ve vztahu k její strategii a okolnímu prostředí. Na druhou stranu tím, že jsou Business Capability realizovány dalšími elementy (lidskými zdroji, systémy, procesy apod.), mohou být popsány, navrženy a implementovány s pomocí přístupů používaných Enterprise Architekty.
- Business Capability se mohou podporovat pomocí vazby Serving.
- Business Capability se popisují obecnými, vysokoúrovňovými termíny, jako je například: Marketing, Customer Contact, Telemarketing, Provide Loans a jsou typicky realizovány kombinací lidských zdrojů, procesy a technickými systémy.
- Business Capability se zpravidla realizují pomocí core elementů, jedním či více. Pro realizaci Business Capability pomocí více elementů, můžeme efektivně využít Grouping.
- Business Capability jsou často využívány při plánování, kdy potřebujeme zachytit jejich vývoj v čase. K tomu můžeme využít prostředky nástroje, pokud je nabízí (Enterprise Architect nabízí několik) nebo je můžeme specializovat, přičemž každá specializace představuje novou verzi (*tak tohle rozhodně nedoporučuji, tohle nedomysleli - PK*). Tyto specializace (a jejich realizace), můžeme agregovat pomocí Plateasu (viz Implementation & Migration) a tím modelovat jejich evoluci.

Elementy v Strategy Extension – Resource #1

Typ: Structure

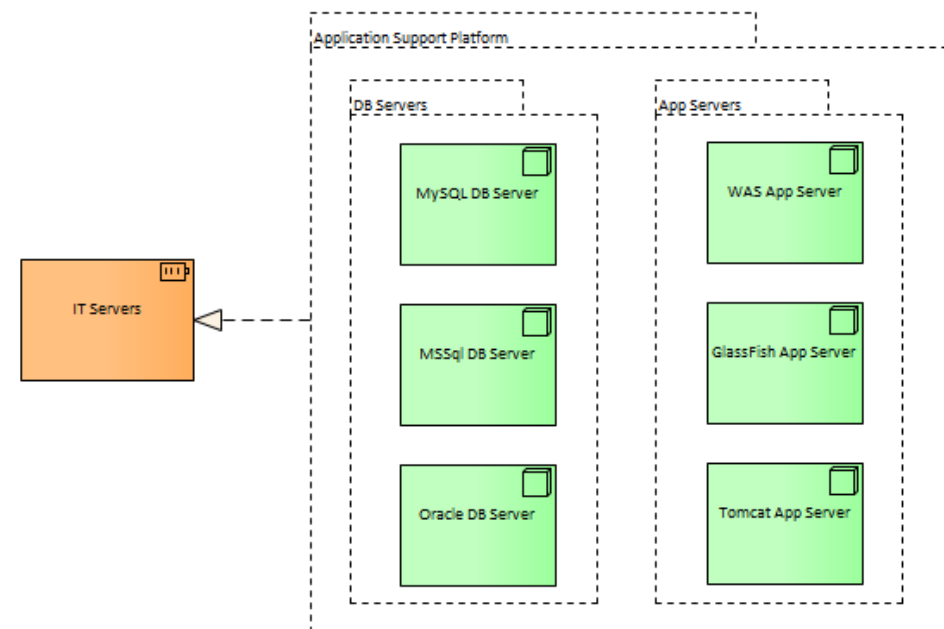


Definice:

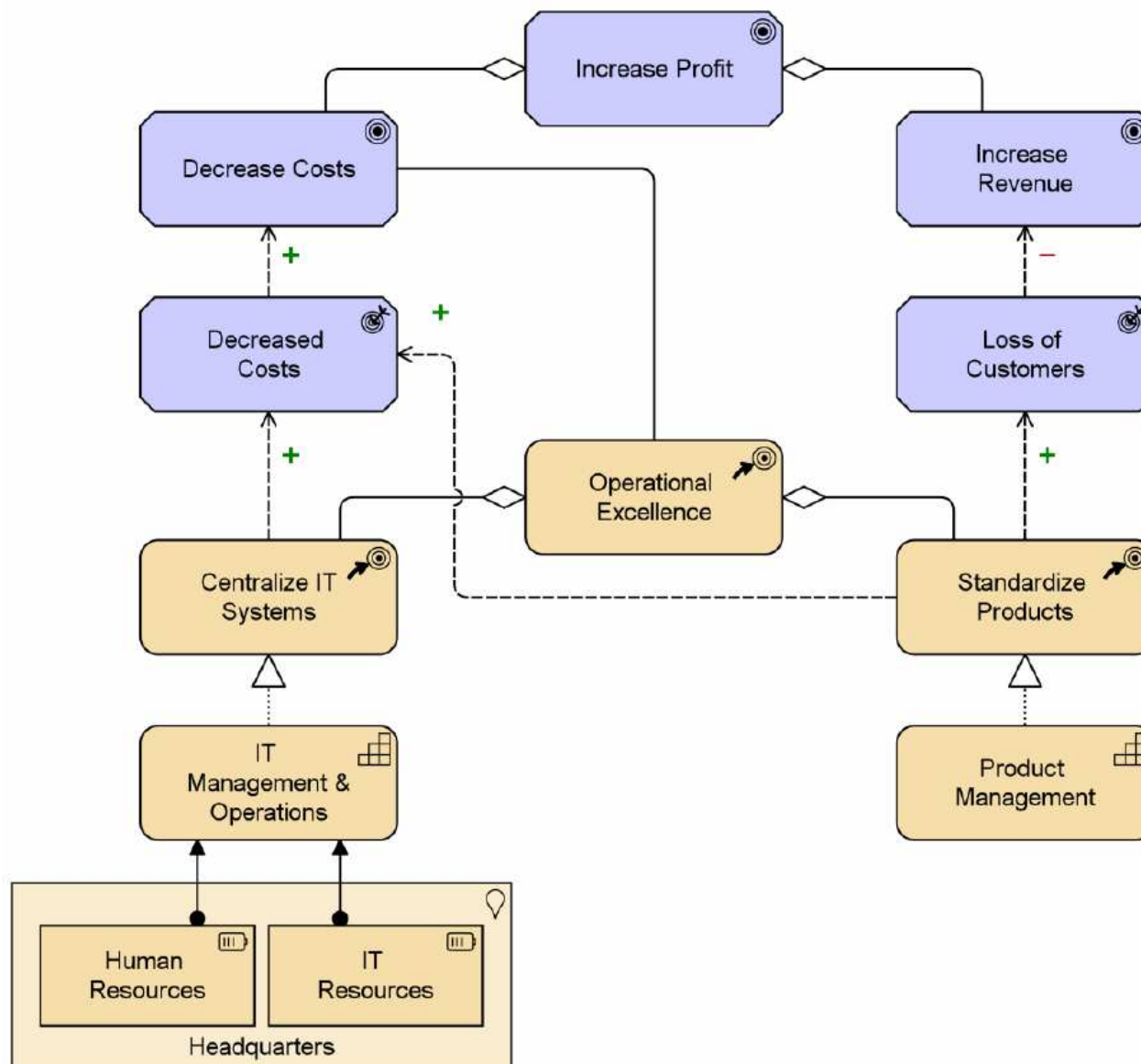
Resource reprezentuje nějaké aktivum, vlastněné nebo ovládané organizací či fyzickou osobou.

Definice :

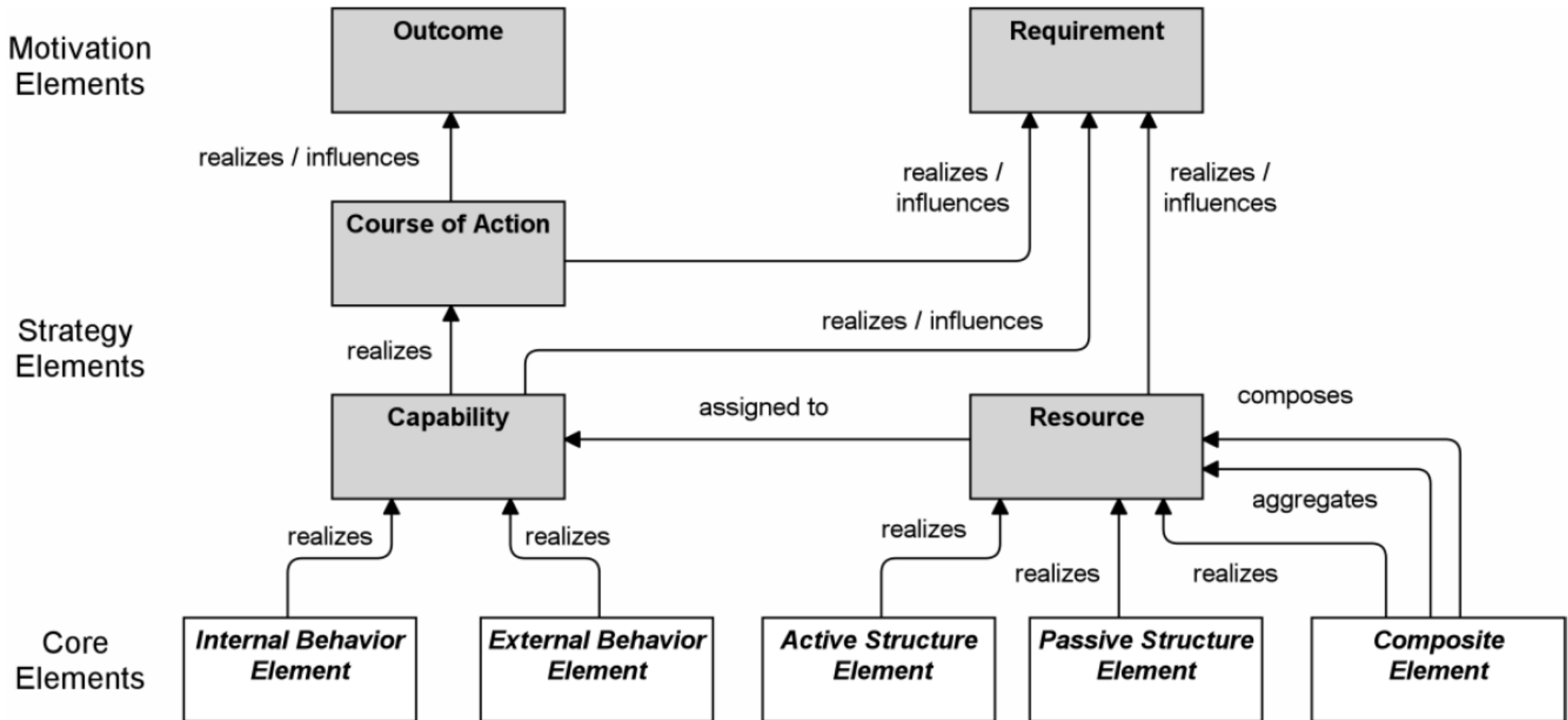
- Resource (zdroje) jsou základním prvkem strategického řízení, ekonomiky, počítačových věd, portfolio managementu a dalších. Velmi často jsou chápány (společně s business capability) jako zdroj konkurenční výhody.
- Resource jsou analyzovány z hledisek silných a slabých stránek a tato analýza je jeden z podkladů pro strategické plánování.
- Protože zdroje (Resource) jsou zpravidla omezené, bývají často rozhodujícím faktorem při určování strategie a cílů organizace a také například, které projekty budou implementovány a které nikoliv (a případně v jakém pořadí).
- Zdroje (Resource) mohou být rozděleny na všeobecné, hmotné (továrny, budovy, vybavení, zásoba minerálů), finanční (peníze, půjčky, akcie, schopnost vypůjčit si peníze), nehmotné (patenty, technologie, práva, patenty, obchodní tajemství, reputace, značka, vztahy s externími entitami, kultura), lidské (znalosti, schopnosti, konkrétní pracovníci, schopnosti spolupráce či komunikace, motivace).
- Resource mohou být realizovány aktivním nebo pasivním strukturálním elementem z jakékoliv vrstvy (Layer).
- Název Resource by měl být podstatné jméno.



Strategy Extension - příklad



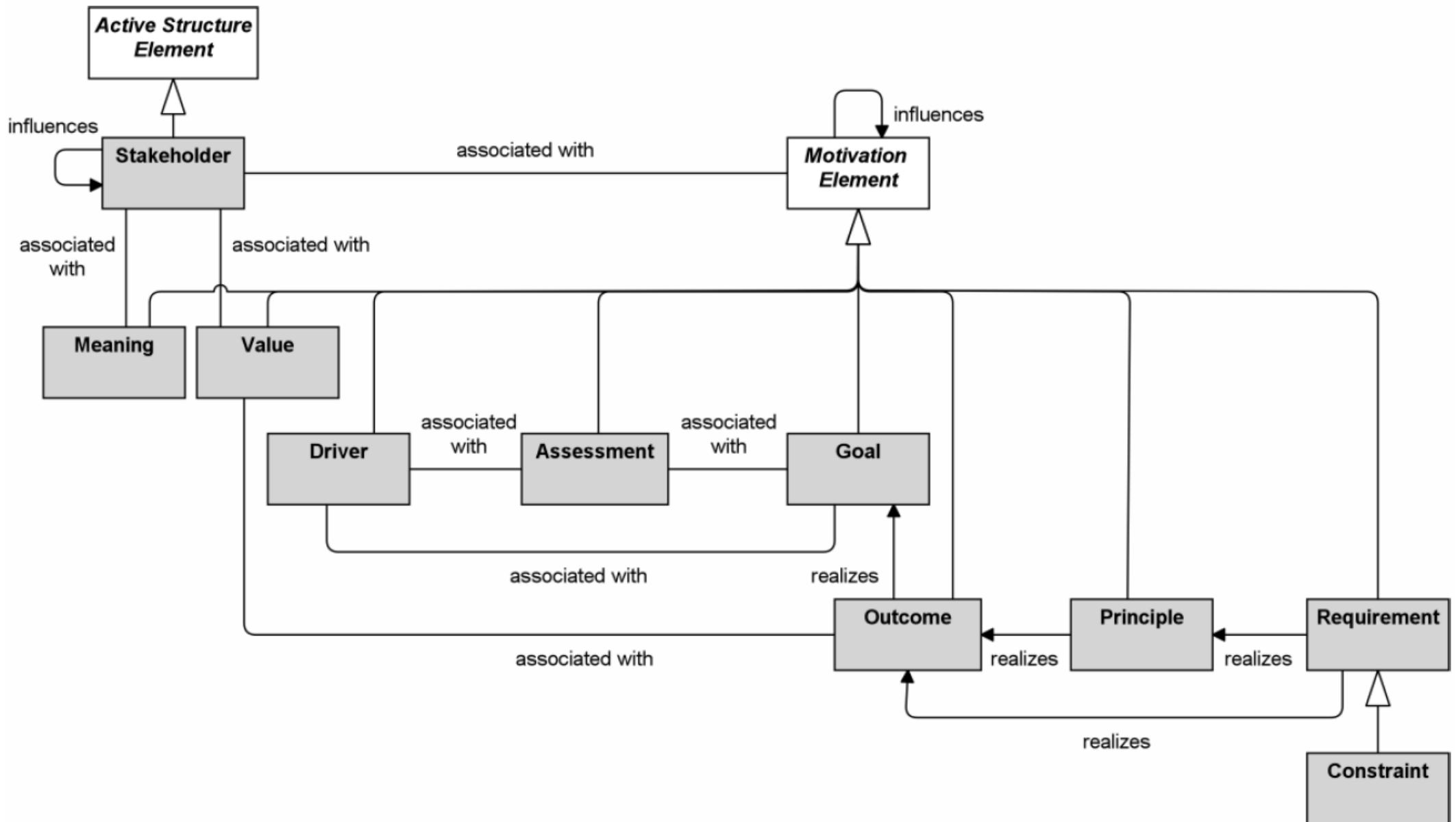
Vztah Strategy a Core elementů - metamodel



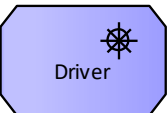
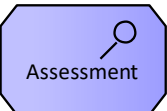
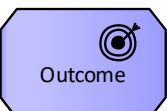
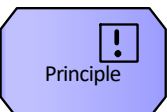
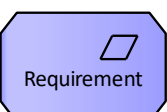
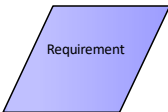
ArchiMate

Elementy – Motivation Extension

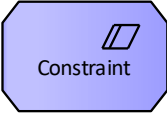
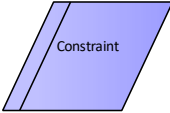
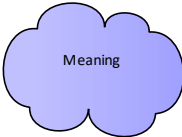
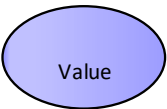
Motivation Extension - metamodel



Elementy v Motivation Extension – přehled #1

Element	Definice	
 Stakeholder	A stakeholder is the role of an individual, team, or organization (or classes thereof) that represents their interests in the outcome of the architecture.	Motivation extension
 Driver	A driver represents an external or internal condition that motivates an organization to define its goals and implement the changes necessary to achieve them.	Motivation extension
 Assessment	An assessment represents the result of an analysis of the state of affairs of the enterprise with respect to some driver.	Motivation extension
 Goal	A goal represents a high-level statement of intent, direction, or desired end state for an organization and its stakeholders.	Motivation extension
 Outcome	An outcome represents an end result that has been achieved.	Motivation extension
 Principle	A principle represents a qualitative statement of intent that should be met by the architecture.	Motivation extension
 Requirement  Requirement	A requirement represents a statement of need that must be met by the architecture.	Motivation extension

Elementy v Motivation Extension – přehled #2

Element	Definice	
 	A constraint represents a factor that prevents or obstructs the realization of goals.	Motivation extension
	Meaning represents the knowledge or expertise present in, or the interpretation given to, a core element in a particular context.	Motivation extension
	Value represents the relative worth, utility, or importance of a core element or an outcome.	Motivation extension

Elementy v Motivation Extension – Stakeholder #1

Typ: Motivation



Definice:

Stakeholder reprezentuje roli , kterou může nabývat fyzická osoba, tým nebo organizace. Tato role reprezentuje jejich zájmy na výstupech architektury.

Definice :

- Element Stakeholder odpovídá definici frameworku TOGAF.
- Stakeholder má jeden či více zájmů v rámci organizace a její enterprise architektury.
- Pro prosazení svých zájmů Stakeholder definuje, mění a zdůrazňuje cíle, kterých má být dosaženo.
- Stakeholder se mohou navzájem ovlivňovat.
- Příklady Stakeholder: CEO, board of directors, akcionáři, zákazníci, business a solution architekti, ale například také státní orgány.



Elementy v Motivation Extension – Driver #1

Typ: Motivation

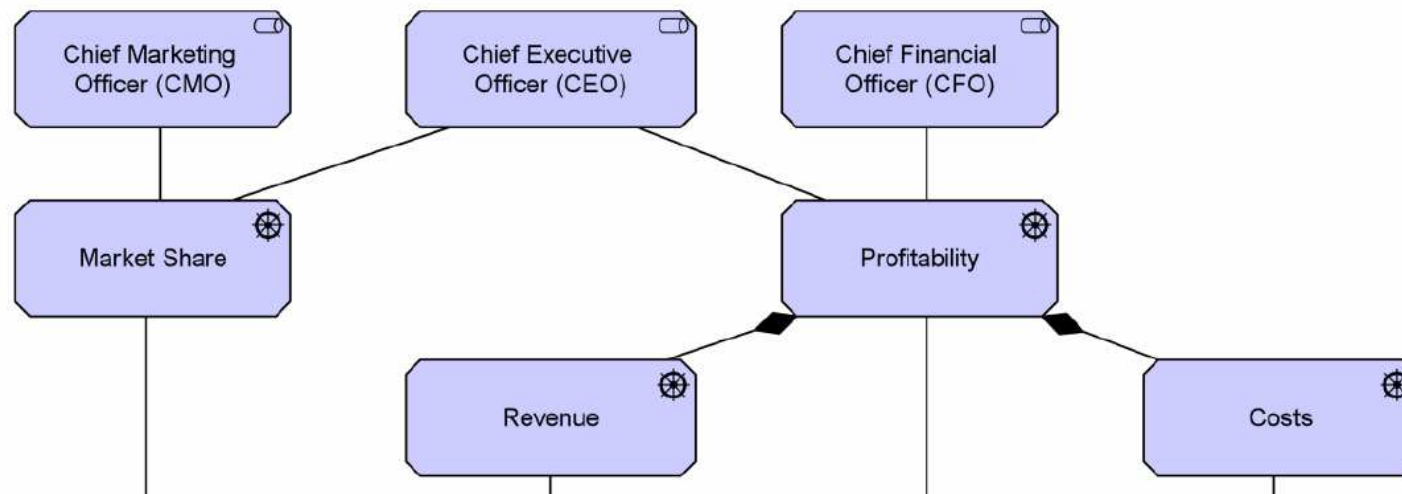


Definice:

Element Driver reprezentuje interní nebo externí podmínku, která motivuje organizaci definovat své cíle a implementovat změny, potřebné pro jejich naplnění.

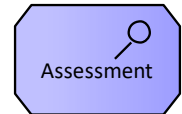
Definice :

- Driver (motivace) mohou být interní a v takovém případě jsou často asociovány (vazba Association) se Stakeholdery. Takové Drivery bývají často nazývány „Concern“ (takto jsou definovány v TOGAF).
- Driver může být prakticky cokoliv, jakákoliv funkčnost, či vlastnost systému (security, vývoj, provoz, výkon, spolehlivost) i cokoliv mimo oblast informačních systémů.
- Další příklady interních motivací (Driver): spokojenost zákazníků, výnosy, snížení nákladů, omezení rizik.
- Příklady externích motivací (Driver): legislativní změny, změny na trhu, kulturní změny.
- Drivery mohou být dekomponovány na podobnější pomocí vazby kompozice (Composition).
- Název driveru by mělo být podstatné jméno.



Elementy v Motivation Extension – Assessment #1

Typ: Motivation

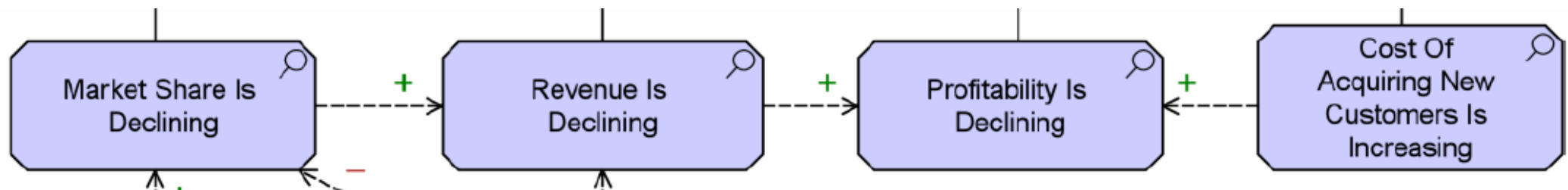


Definice:

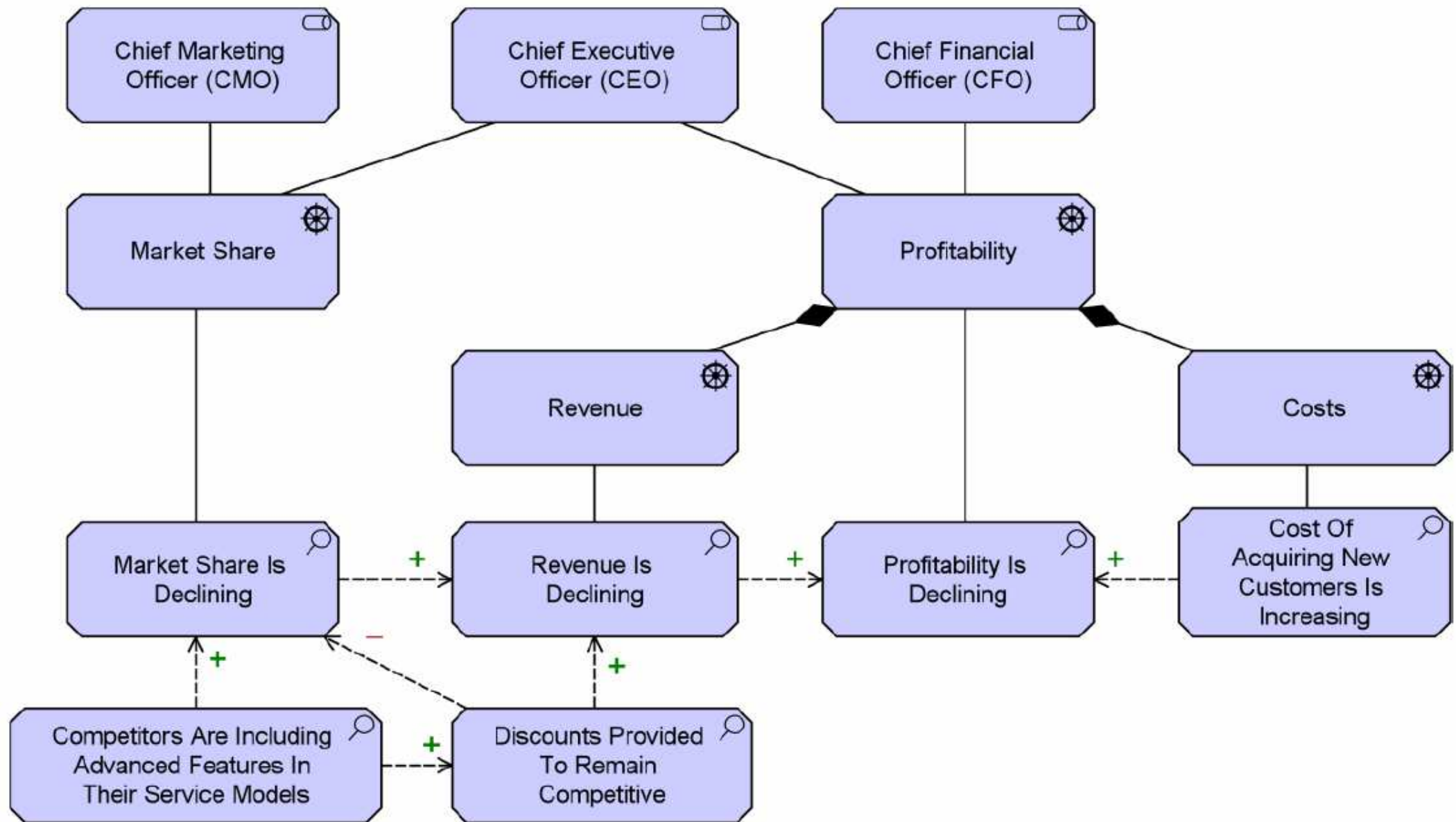
Assessment představuje výsledek analýzy aktuálního stavu věcí uvnitř organizace se zřetelem na některé Drivery.

Definice :

- Assessment může odhalit silné a slabé stránky společnosti, příležitosti a nebo hrozby (v určité oblasti zájmů). Assessmenty by měly být adresovány (pokud je to potřeba) změnou existujících cílů nebo nastavením nových a tyto cíle mohou spouštět změny Enterprise Architektury.
- Assessment může být chápán jako tvrzení.
- Silné a slabé stránky jsou z hlediska organizace interní faktory, příležitosti (Opportunities) a hrozby (Threats) jsou z hlediska organizace externí.
- Slabé stránky a hrozby musí být považovány za problém, a tedy musí být cíle nastaveny tak, aby tyto problémy eliminovaly.
- Silné stránky a příležitosti mohou být většinou přímo transformovány do cílů.
- Příklad: výstup Assessmentu s názvem: „Zákazník si stěžuje na použitelnost helpdesku“ může být adresován cílem „Vylepšit helpdesk“. Nebo, příležitost (Opportunity): „Zákazníci rádi spravují své portfolio online“ může být transformována do cíle „Zavést online portfolio management“.
- Assessment mohou být dekomponovány pomocí vazby „Composition“.
- Assessment se navzájem mohou ovlivňovat pomocí vazby Influence.
- Assessment by měl mít v názvu podstatné jméno nebo velmi krátkou větu.



Motivation Extension – příklad #1



Elementy v Motivation Extension – Goal #1

Typ: Motivation

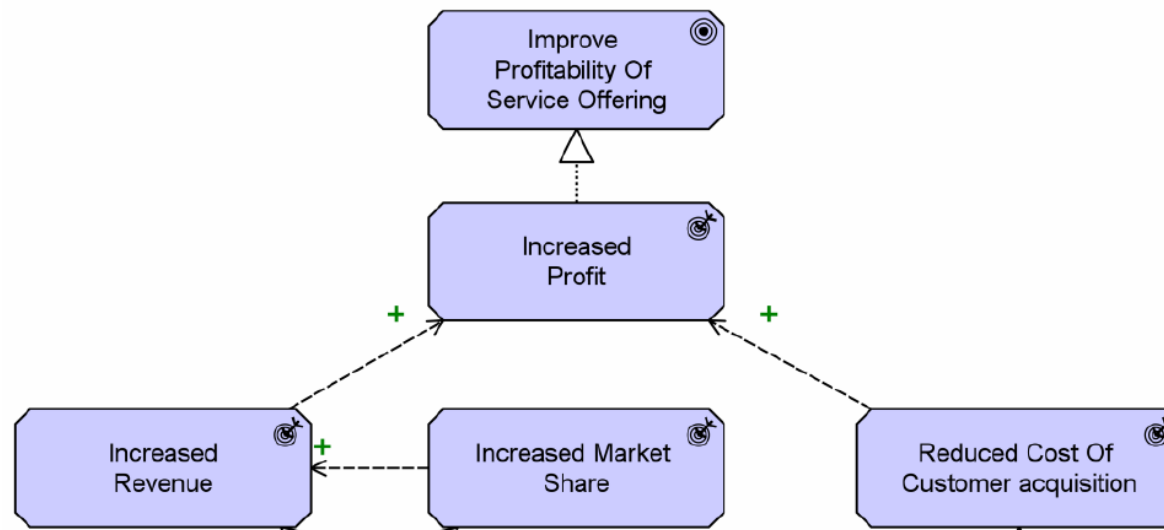


Definice:

Goal reprezentuje vyjádření záměru (na vysoké úrovni), směřování nebo požadovaného cílového stavu organizace a jejich stakeholderů.

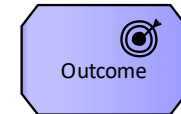
Definice :

- Goal může reprezentovat cokoli, co si stakeholderi mohou přát, ať se již jedná o cílový stav něčeho nebo nějakou vyprodukovanou hodnotu. Příklady: zvýšení zisku, snížení doby čekání na helpdesku nebo zavedení online portfolio managementu.
- Goal (cíle) se typicky používají pro měření úspěchu v organizaci (respektive jejich plnění).
- Při definici cílů (Goal) se často používají výrazy jako: zvýšit, vylepšit, zjednodušit apod.
- Cíle (Goal) se mohou dekomponovat (vazby Aggregate a Composite). Například cíl Zvýšení zisku se může dekomponovat na Zvýšení příjmů a Snížení nákladů.
- Velmi často se cíli (Goal) přiřazuje (pomocí realizace) konkrétní výstup (Outcome), který popisuje přesně co a kdy má být dodáno pro splnění cíle (Goal).



Elementy v Motivation Extension – Outcome #1

Typ: Motivation

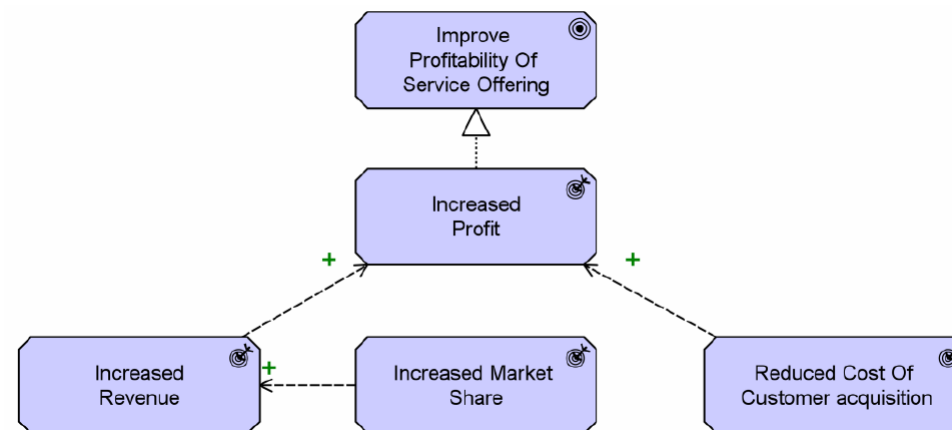


Definice:

Outcome reprezentuje konkrétní výstup či výsledek, jehož musí být dosaženo.

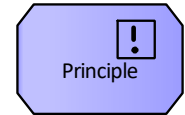
Definice :

- Outcome je výsledek či výstup, definovaný na vysoké úrovni, orientován na business a je produkován schopnostmi (Capability) organizace (a činností core element, které tyto capabilities realizují).
- Outcome (výstupy) jsou hmatatelné, často kvantitativně vyjádřitelné, vztažené k času a mohou být navázány na Assesmenty.
- Různé výstupy (Outcome) mohou mít pro různé stakeholdery odlišnou hodnotu.
- Výstupy (Outcome) jsou důležité pro outcome-oriented přístup k plánování Enterprise Architektury a také pro plánování založeném na schopnostech organizace (Capability-based planning).
- Výstupy (Outcome) mají blízký vztah k požadavkům (Requirement), cílům (Goal) a dalším elementům vyjadřujícím záměry.
- Výstupy (Outcome) jsou konečné výsledky - a požadavky a cíle jsou často formulovány právě jejich prostřednictvím (tedy, cíl je splněn, pokud bylo dosaženo určitého výstupu – Outcome).
- Výstupy (Outcome) mohou být dekomponovány pomocí Aggregation nebo Composition.
- Názvy výstupů (Outcome) by měly obsahovat podstatné jméno a vyjádření, že výsledku bylo dosaženo (v minulém tvaru):
Nový systém byl implementován, Zisk za poslední čtvrtletí se zvednul o 15% procent, Bylo ustanoveno klíčové partnerství s dodavatelem. Nebo více specificky, Zisk za Q4/2017 dosáhl hranice 100 milionů korun.



Elementy v Motivation Extension – Principle #1

Typ: Motivation



Definice:

Principle reprezentuje vyjádření záměru, kterého by mělo být dosaženo.

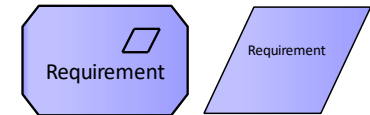
Definice :

- Principy jsou velmi podobné cílům (Goal) a požadavkům (Requirement).
- Na rozdíl od požadavků a cílů jsou principy (Principle) více abstraktní a mají širší záběr.
- Principy (Principle) definují obecné požadavky, které má splňovat každý systém v rámci architektury, či její oblasti (definuje se v rámci principu), požadavky (Requirement) se definují konkrétně a pro konkrétní systém v rámci architektury.
- Principy (Principle) potřebují být dále specifikovány pro konkrétní systémy (jak jich má být dosaženo) a to se provádí prostřednictvím jednoho či více požadavků (požadavky realizují principy). Například princip, „Procesy pro řízení IT musí odpovídat všem platným zákonům, pravidlům a regulacím“ bude (na konkrétním systému) realizován požadavky (Requirements), které přesně definují jaké zákony a jakým způsobem se budou aplikovat.
- Principy (Principle) jsou motivovány cíly (Goal) a Drivery. Například, v předchozím bodě zmíněný princip může být motivován cílem „Udržení dobré pověsti“ anebo cílem „Vyhnutí se pokutám“.
- Principy (Principle) bývají někdy označovány jako Guidelines.
- Principy mohou realizovat cíle (Goal) a tím nastavit základní pravidla pro všechny systémy (v zadané oblasti či kontextu), která musí být dodržena, aby cíl byl splněn.
- Principy (Principle), na rozdíl od požadavků, bývají stabilní v čase a nemění se tak často.
- Principy (Principle) mohou zohledňovat hodnoty organizace, best practices, znalosti atd. ve svých vyjádřeních.
- Principy (Principle) mohou být dekomponovány pomocí Aggregation nebo Composition.
- Název by měl obsahovat krátký souhrn principu.



Elementy v Motivation Extension – Requirement #1

Typ: Motivation

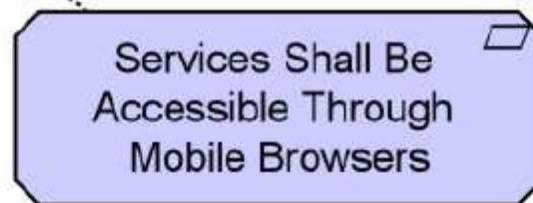


Definice:

Požadavek (Requirement) reprezentuje konkrétní potřebu, která musí být naplněna.

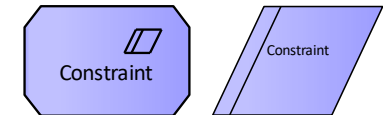
Definice :

- Požadavek (Requirement) je prostředek k naplnění cílů (Goal). Tyto cíle jsou požadavky realizovány.
- Požadavek (Requirement) se zpravidla vztahuje k systému (čímž se rozumí jakýkoliv core element vztahující se k organizaci), tedy Business actor, process, Application Component, Application Service apod.).
- Požadavek (Requirement) definuje, co přesně má být v určitém systému, aplikaci či business entitě vykonáno.
- Každý cíl (Goal) může být zpravidla realizován několika způsoby. Zvolený způsob je právě definován požadavkem.
Například cíl (Goal): Improve Portofolio Management může být řešen požadavkem „Přidělit každému zákazníkovi osobního asistenta“ nebo požadavkem „Implementovat novou funkčnost Online Portofilo Management“.
- Požadavky se mohou dále dekomponovat na konkrétní funkční požadavky „Implementovat formulář do systému“, „Evidovat nová data“ apod.
- *Existuje několik metodik jak psát správně požadavky, mezi známější patří například metodika SMART.*



Elementy v Motivation Extension – Constraint #1

Typ: Motivation

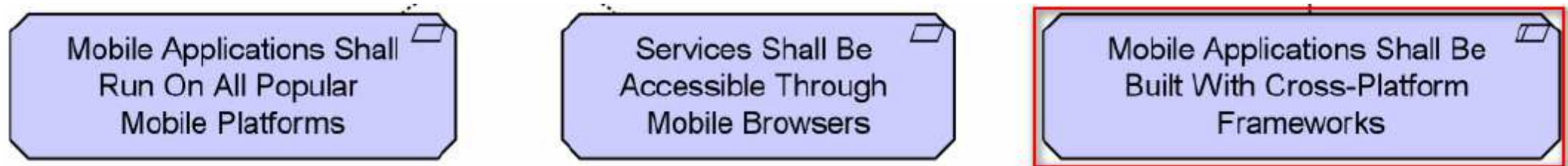


Definice:

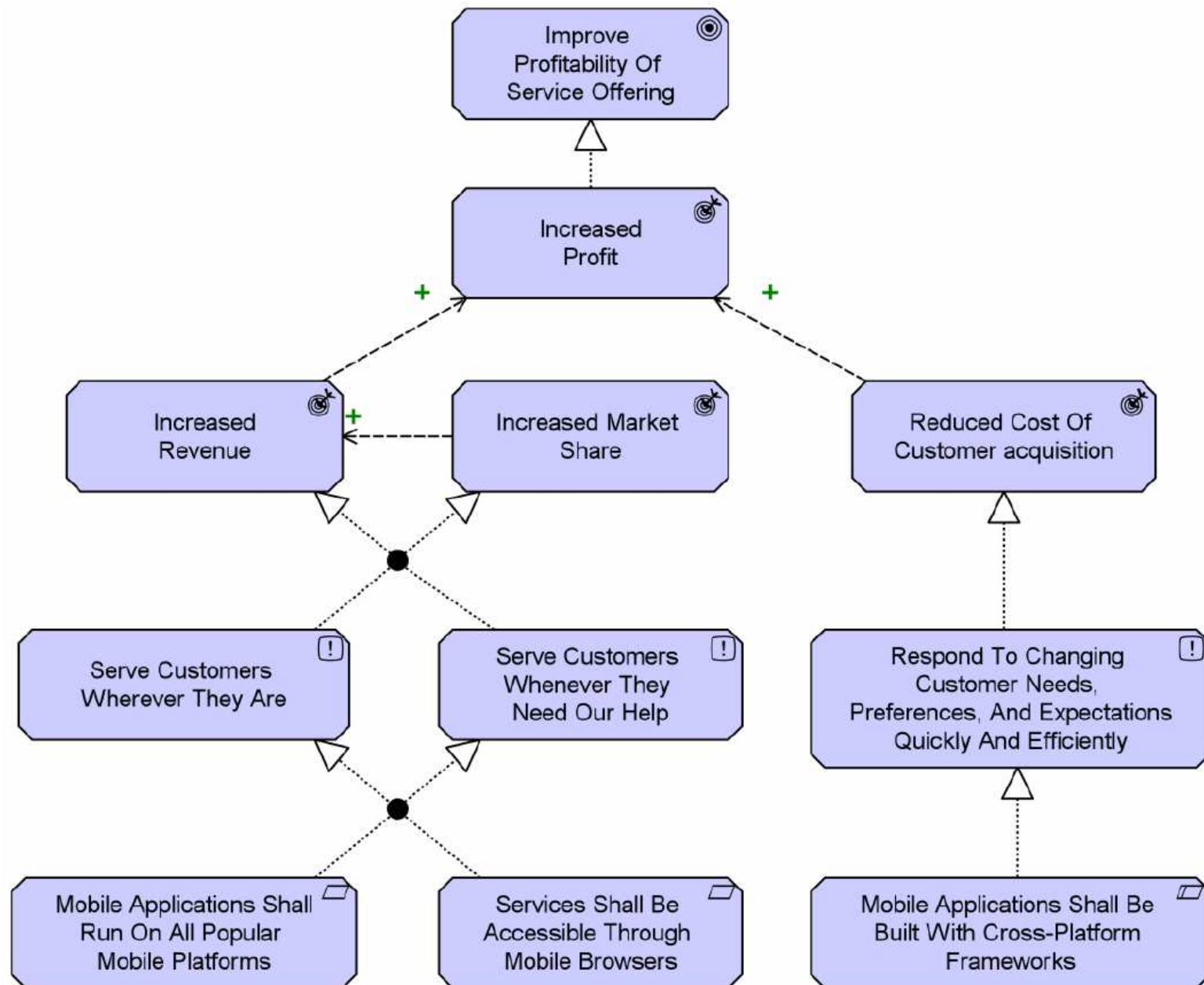
Constraint (omezení) reprezentuje nějaký faktor, který brání nebo komplikuje dosažení určitého cíle.

Definice :

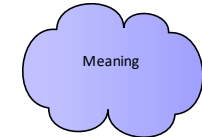
- Na rozdíl od požadavku (Requirement) nedefinuje omezení (Constraint) nové funkčnosti, ale naopak stanovuje omezení při realizaci požadavků.
- Tato omezení (Constraint) se mohou týkat implementace nebo provozování systému, mohou být technického, právního, finančního (omezený rozpočet) či jiného charakteru. Často se omezení týká omezení zdrojů.
- Omezení (Constraint) mohou být dekomponovány pomocí Aggregation nebo Composition.
- Omezení (Constraint) mohou být připojena k core elementům.



Motivation Extension – příklad #2



Elementy v Motivation Extension – Meaning #1



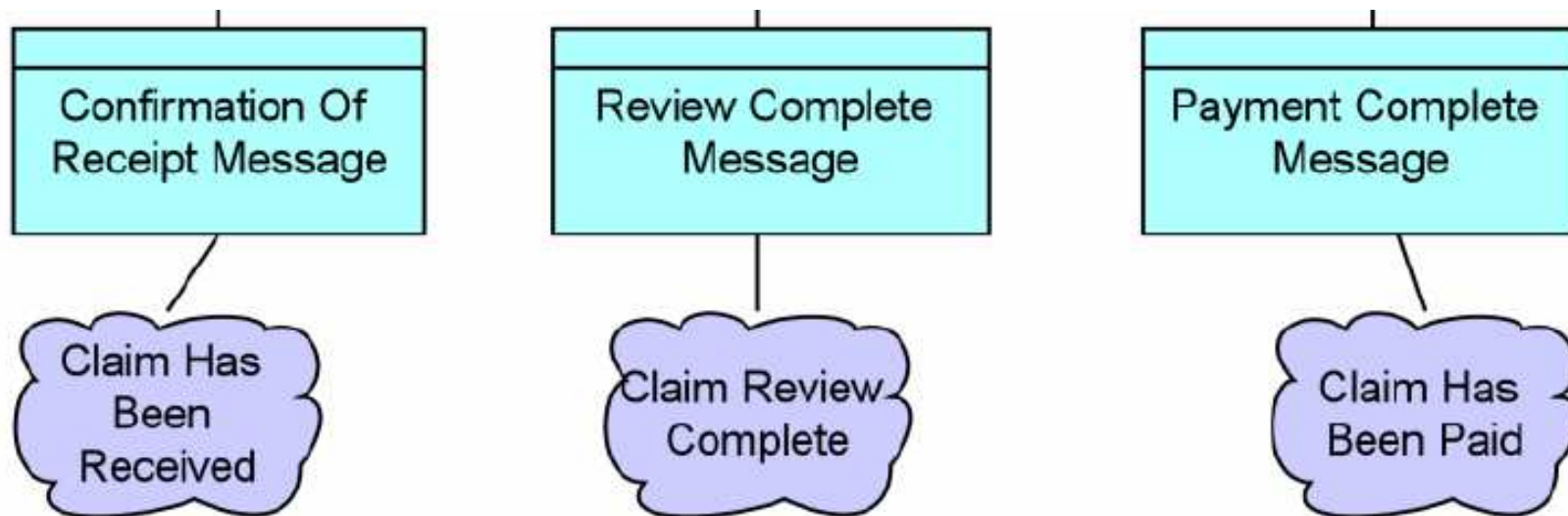
Typ: Motivation

Definice:

Meaning (význam) reprezentuje nějakou znalost či zkušenost. Dále reprezentuje konkrétní interpretaci danou core elementu v určitém kontextu.

Definice :

- Element Meaning interpretuje element uvnitř architektury. Velmi často se používá pro popis významu pasivních struktur (passive structure element), například dokumentů či zpráv.
- Různí uživatelé mohou vnímat různé funkcionality, data, dokumenty a jiné elementy odlišně.
- Meaning (význam) může být přiřazen (pomocí Association) jakémukoliv core elementu. Pro znázornění odlišného významu pro různé stakeholdery, může být Meaning (význam) zároveň přiřazen příslušnému stakeholderovi.



Elementy v Motivation Extension – Value #1

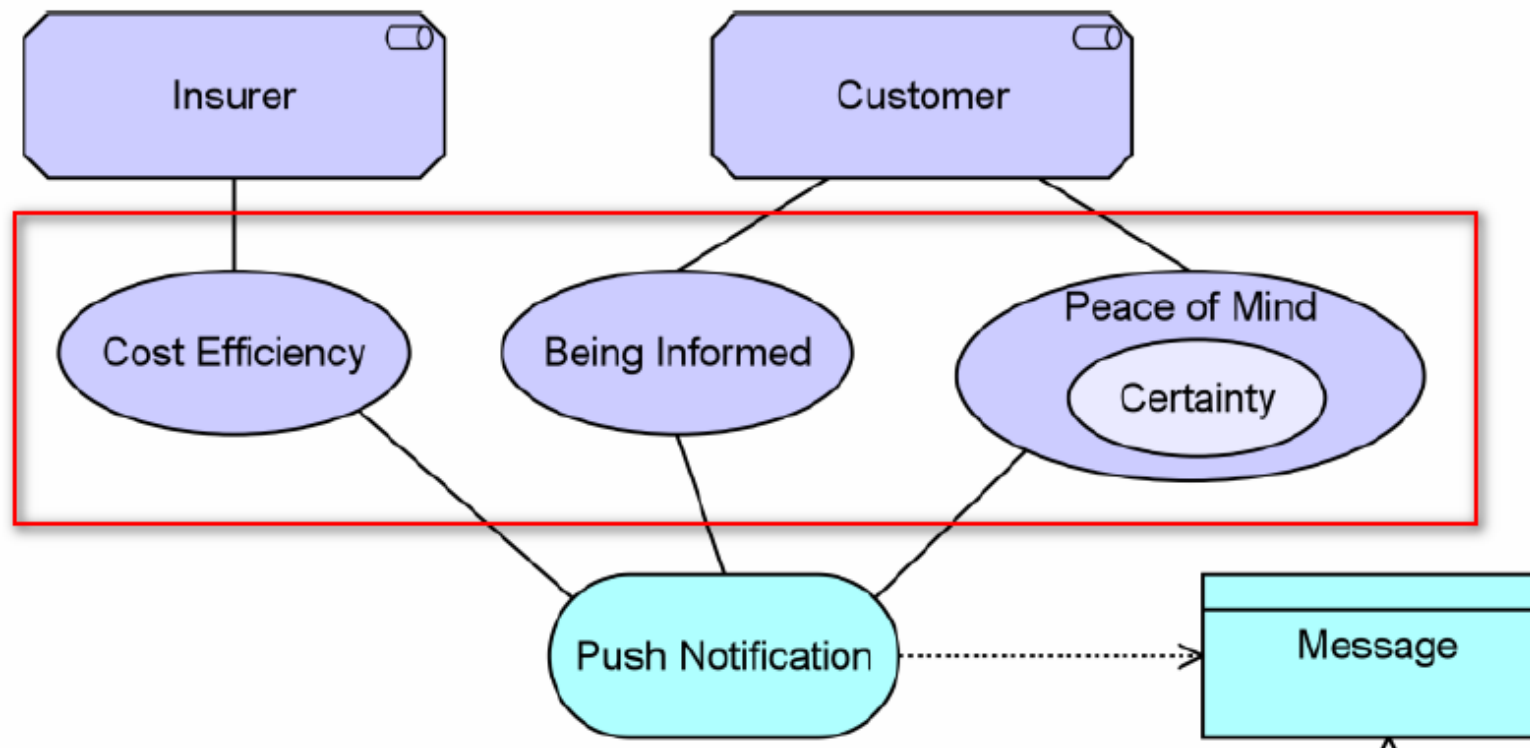
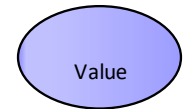
Typ: Motivation

Definice:

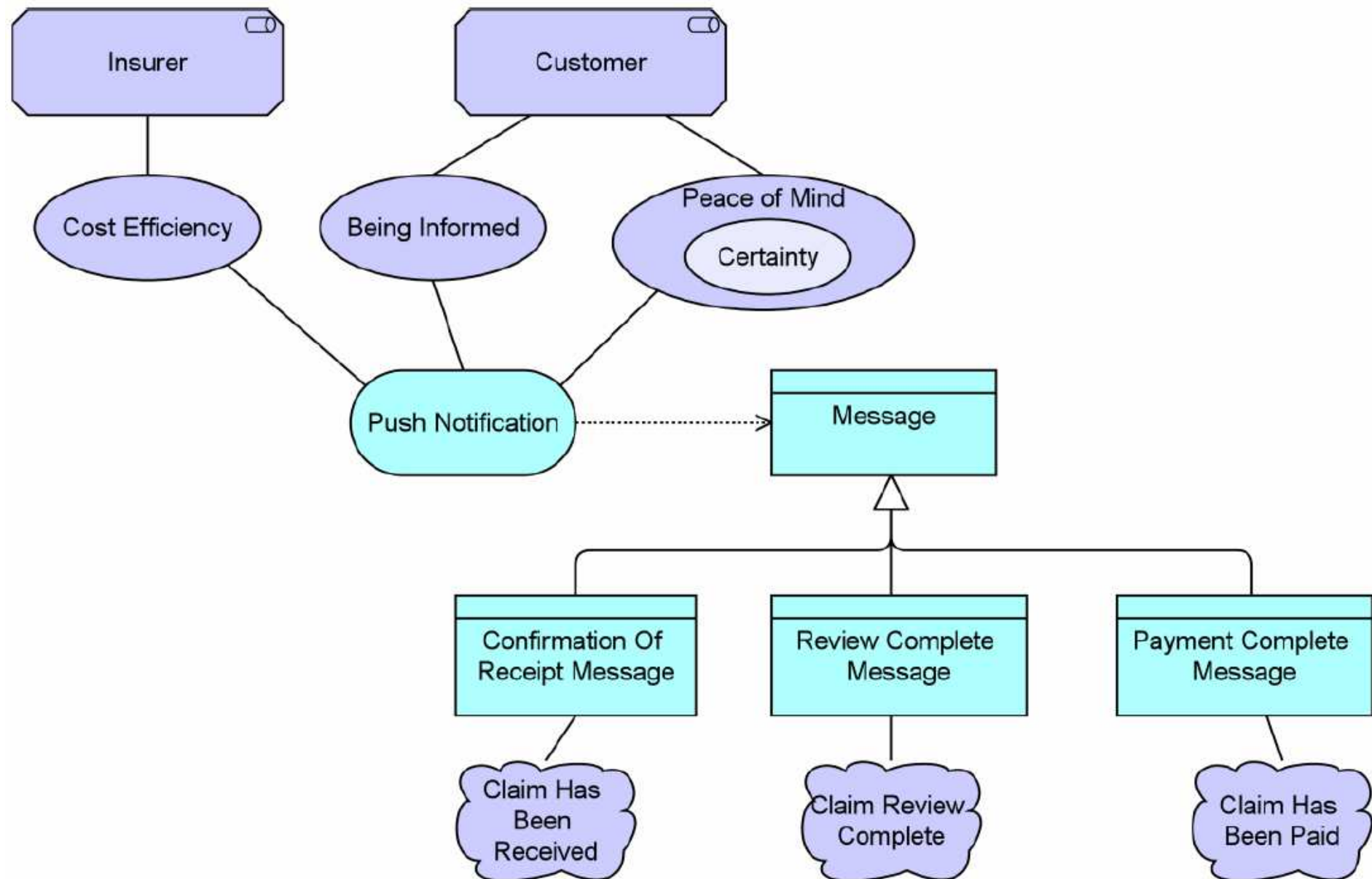
Value (hodnota) reprezentuje relativní hodnotu nebo důležitost core elementu nebo výstupu (Outcome).

Definice :

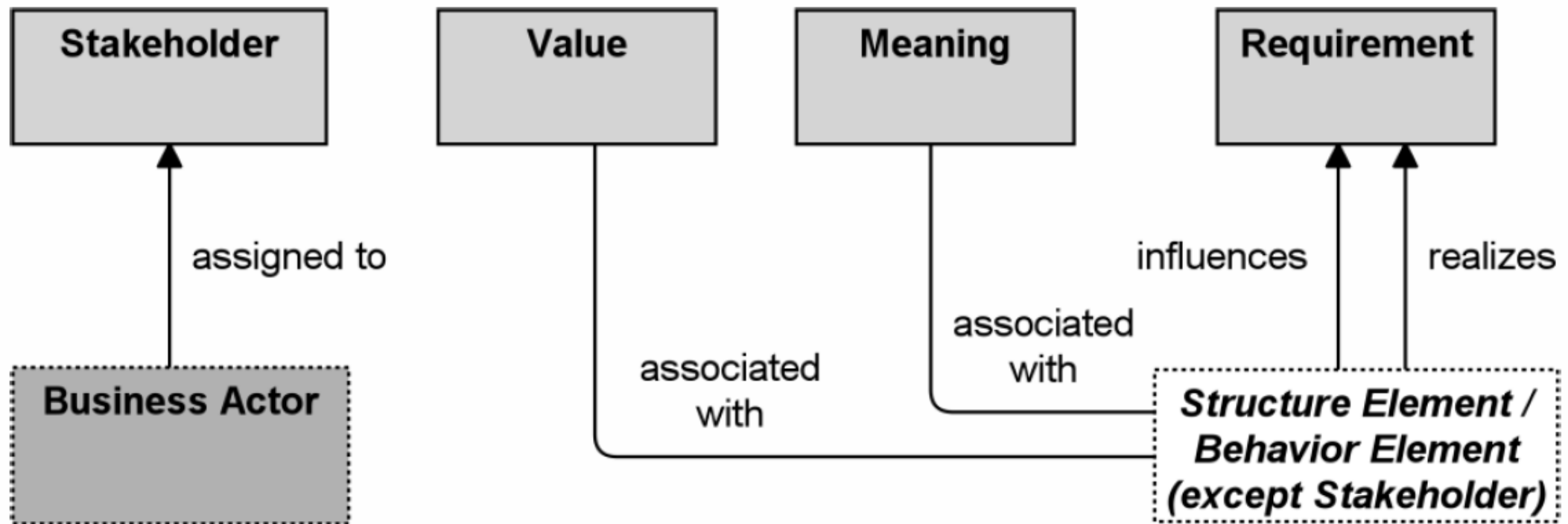
- Value je často vyjádřena v penězích.
- Value může být přiřazena (pomocí vazby Association) k libovolnému core elementu a jeho výstupům.
- Pro namodelování, že Value může zajímat konkrétního stakeholdera, může s ním být rovněž asociována.
- Název Value může být vyjádřen různě, včetně zahrnutí množství, spokojenosti zákazníka apod.



Motivation Extension – příklad #3



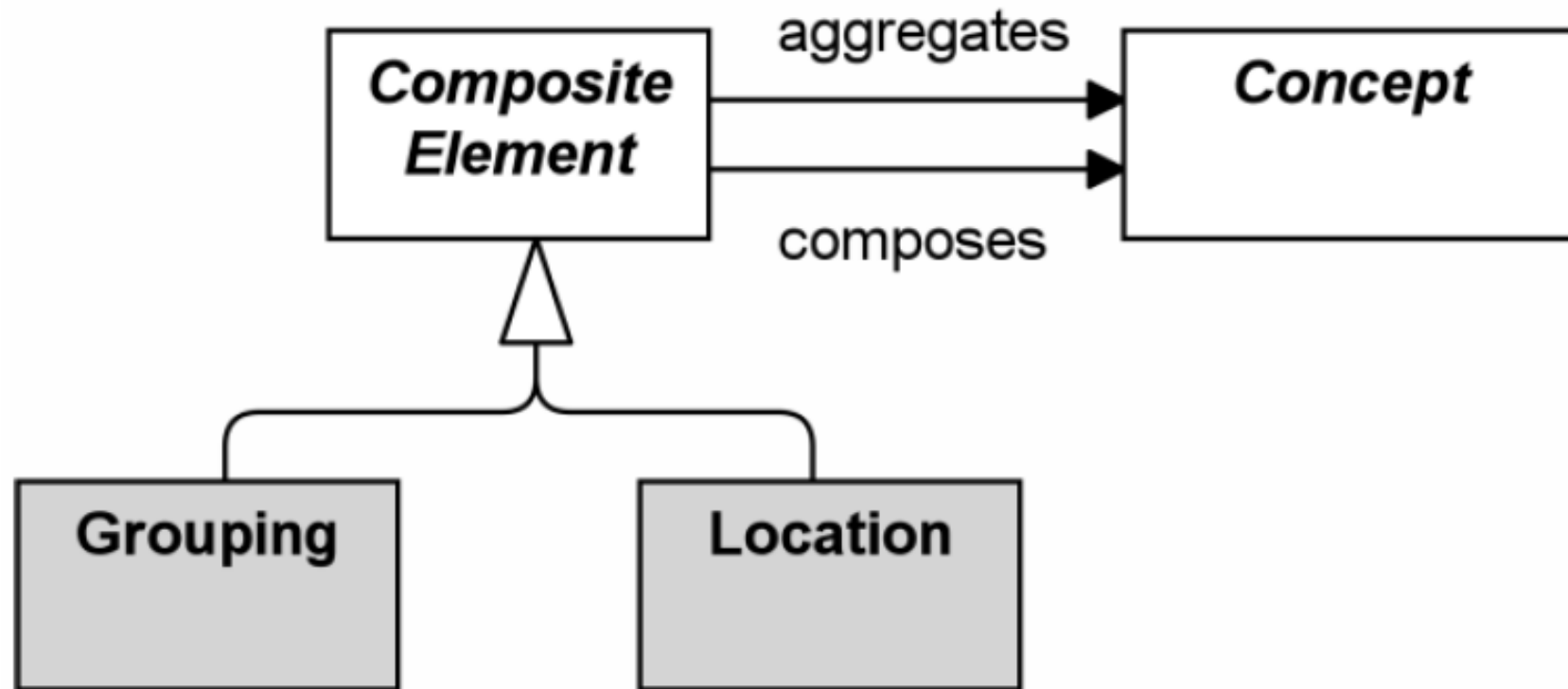
Vztah Motivation a Core elementů - metamodel



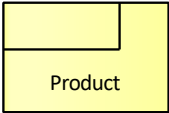
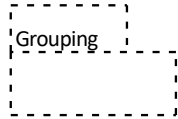
ArchiMate

Elementy – Composite

Kompozitní elementy - metamodel

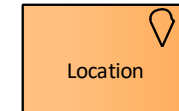


Elementy typu Composite – přehled #1

Element	Definice	
	A location is a place or position where structure elements can be located or behavior can be performed.	Composite
	A product represents a coherent collection of services and/or passive structure elements, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.	Composite
	The grouping element aggregates or composes concepts that belong together based on some common characteristic.	Composite

Composite elementy– Location #1

Typ: kompozitní (Composite)

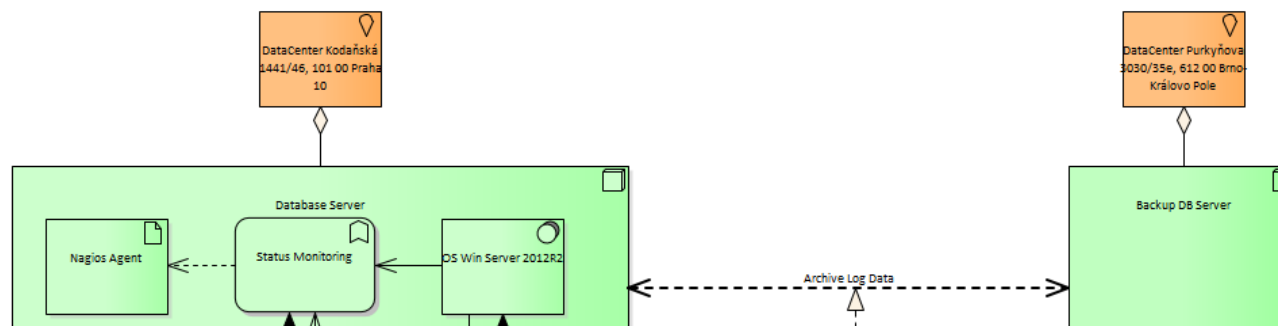


Definice:

Location je místo nebo pozice, kde můžou být umístěny strukturální elementy nebo kde může být vykonáváno chování (behavior)

Definice :

- Location element je použit k namodelování, kde se nachází (aktivní nebo pasivní) strukturální elementy, jako je například Business Actor, Application Component, Device, Node apod. Nebo elementy typu Behavior, čímž se znázorní, kde je dané chování vykonáváno.
- Location je element typu Grouping - všechny prvky, uvedené v předchozím bodě se do něj agregují a to buď přímým umístěním do elementu Location nebo napojením pomocí vazby Aggregation (diamond je na straně Location).
- Location může být dále dekomponováno pomocí vazby Aggregation nebo Composition.
- Location (a tedy její název) může být konkrétní (adresa, umístění v serverovně apod.) nebo relativní – Primary Datacenter / Backup Datacenter apod.
- *Location koresponduje se sloupečkem Where v Zachman Framework.*



Composite elementy– Product #1

Typ: kompozitní (Composite)

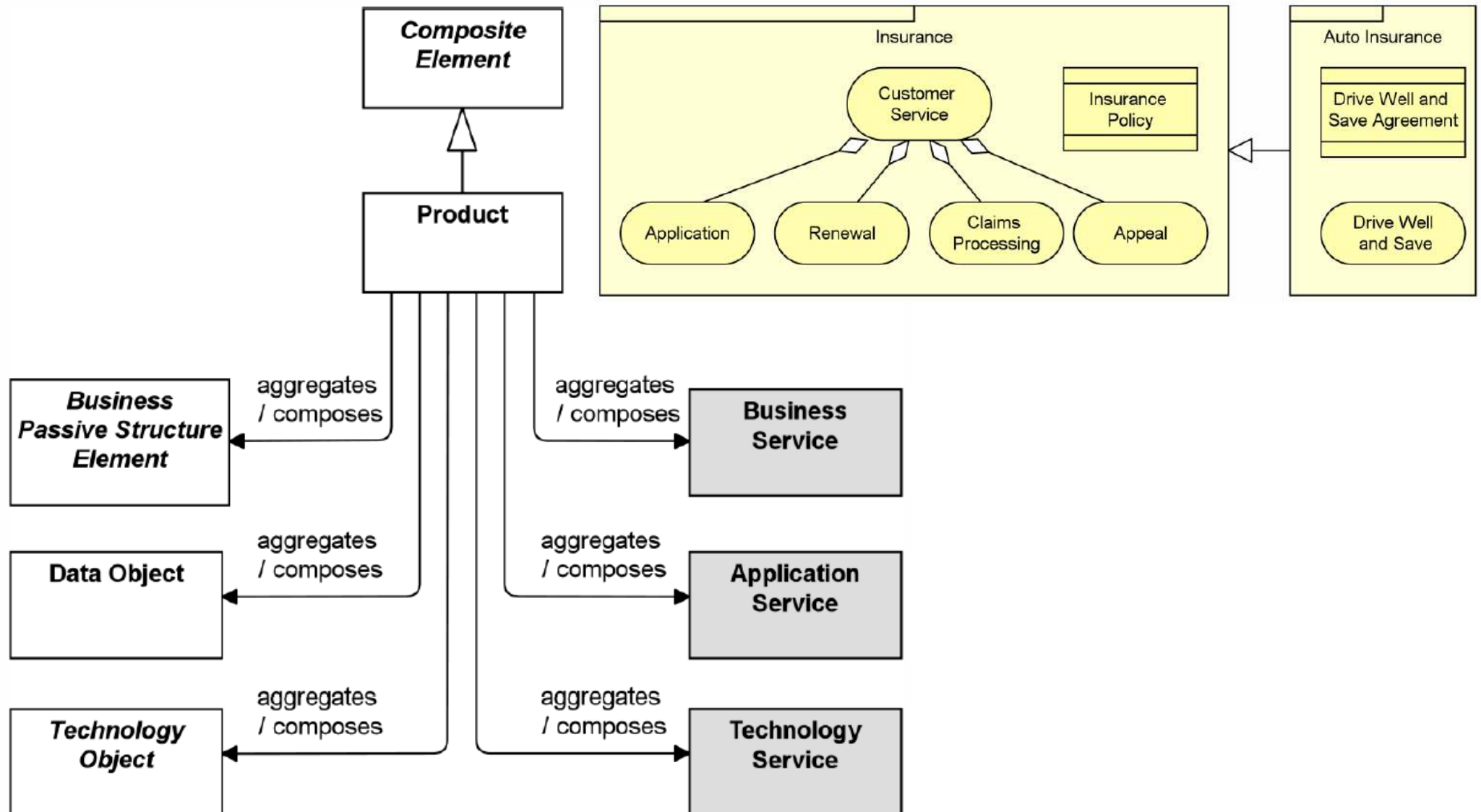
Definice:

Produkt (Product) reprezentuje koherentní soubor služeb (Business Service) a/nebo pasivních struktur (Passive Structure) doprovázených jedním (či více) kontraktem. Produkt je nabízen interním či externím klientům jako celek.

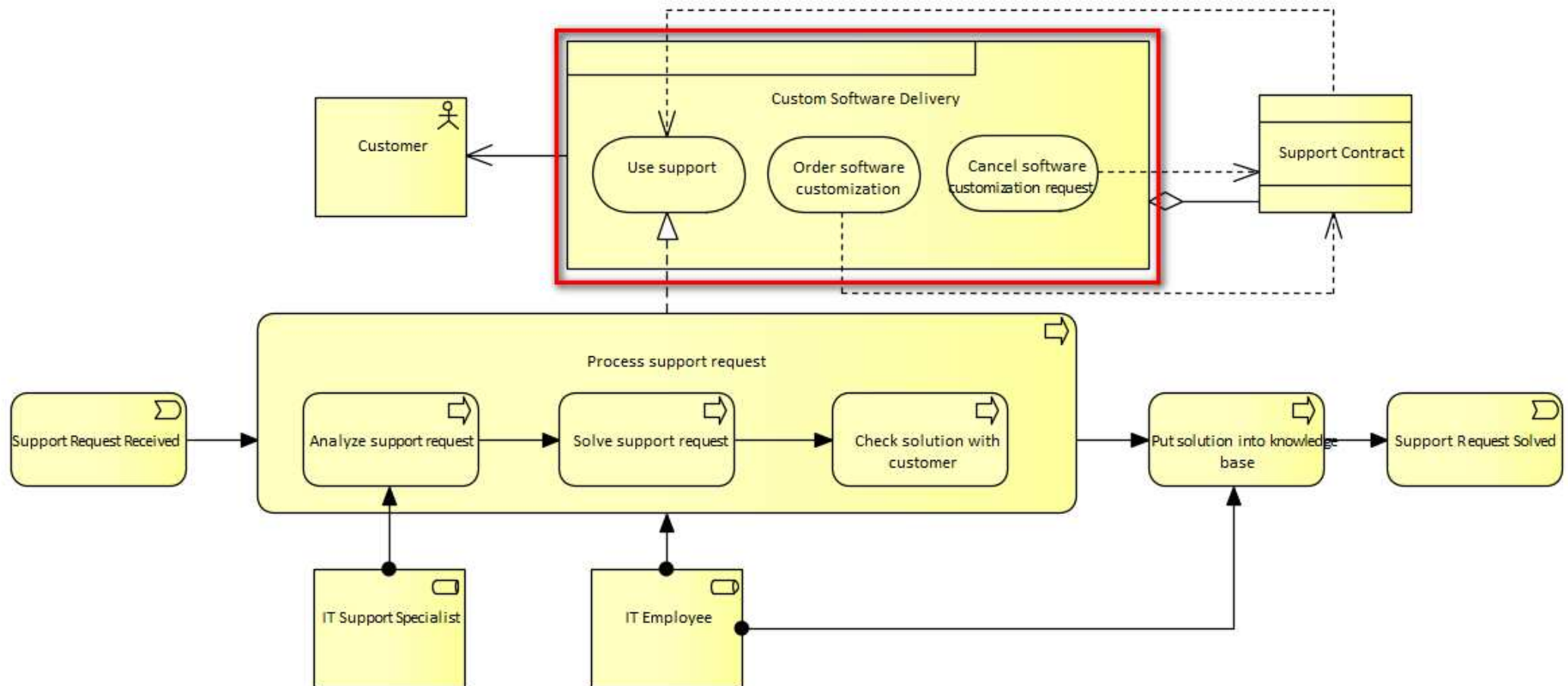
Definice :

- Výše uvedená definice pokrývá fyzické produkty i produkty založené na službách či informacích.
- Zakoupení, či jiné získání produktu zákazníkem, mu dává právo využívat souvisejících služeb (zahrnutých v produktu).
- Element Product je chápán jako specifikace **typu** produktu (tedy nikoliv jako konkrétní instanci produktu). V běžné organizaci zpravidla bývá počet typů produktů omezen a je stabilní, na rozdíl od business procesů, funkcí apod.
- Pro zakoupení, či získání produktů zákazníkem bývá zpravidla definována business služba (Business Service), která způsobí vytvoření „instance“ produktu. A obdobně jiná služba slouží pro její ukončení.
- Produkt může zahrnovat elementy z business vrstvy, ale i z jiných.
- S produktem může být spojena Value (viz další výklad).
- Název produktu je obvykle stejný, jaký je komunikován interním či externím zákazníkům.

Composite elementy– Product #2



Composite elementy– Product #3



Composite elementy– Grouping #1

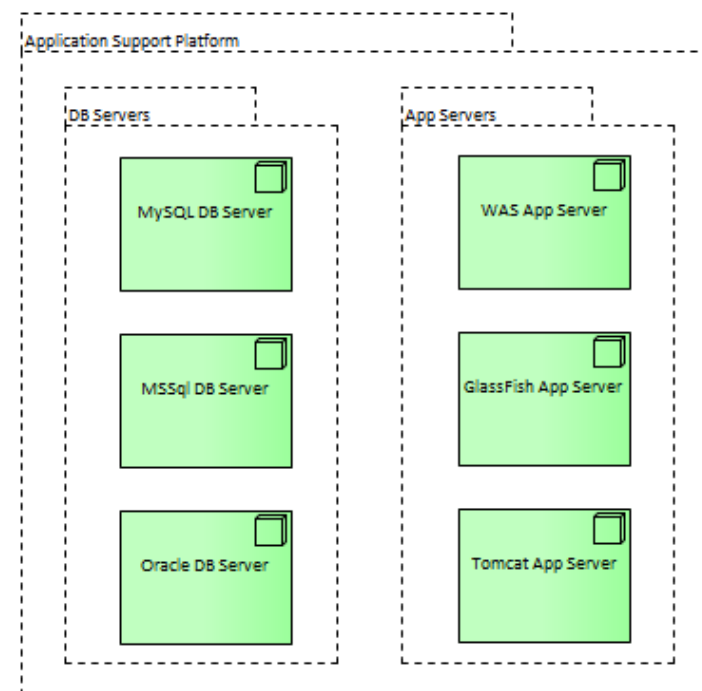
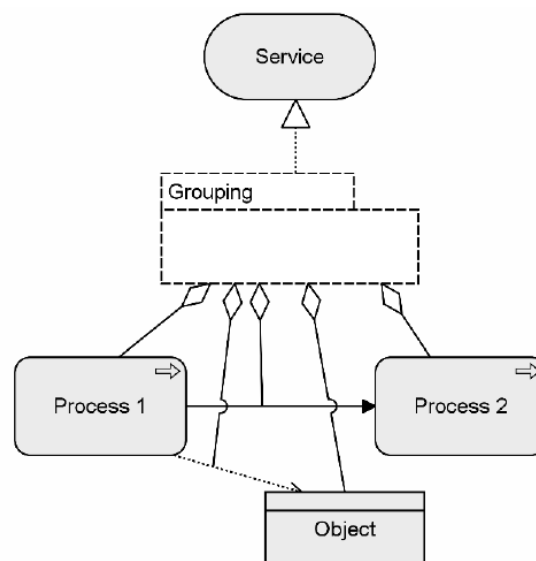
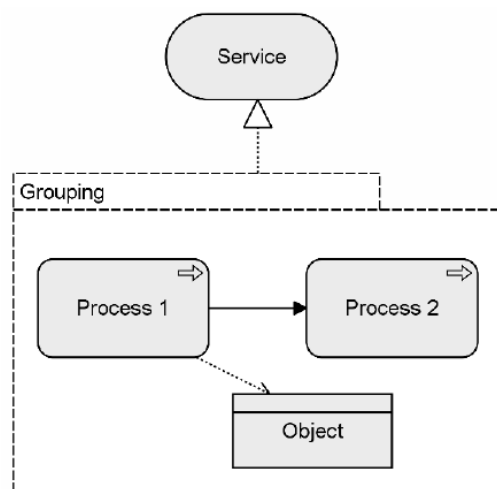
Typ: kompozitní (Composite)

Definition:

Element Grouping seskupuje vazby (Aggregation nebo Composition), které spolu souvisí na základě nějaké sdílené charakteristiky.

Definition :

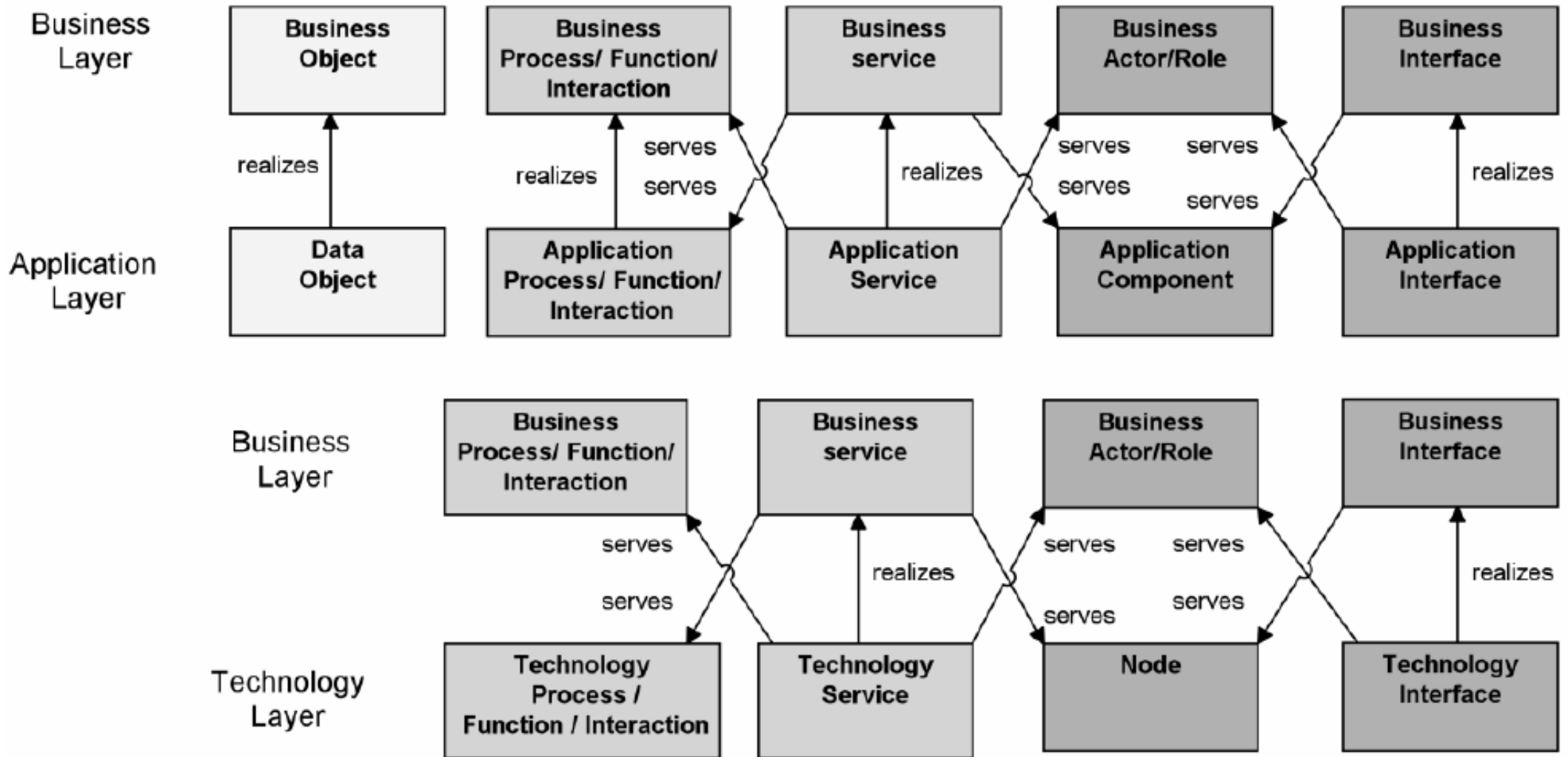
- Grouping slouží k seskupování libovolných konceptů (vazeb a elementů). Používají se k tomu buď vazby (Aggregation nebo Composition) nebo tím, že se koncepty umístí přímo do elementu grouping.
- Grouping elementy se mohou také navzájem seskupovat (Aggregation nebo Composition) nebo pokud modelujeme přímo do elementu Grouping, tak navzájem překrývat.
- Grouping slouží také k vytváření Architecture a Solution Building Block (ABBs a SBBs) tak, jak jsou definovány ve frameworku TOGAF.



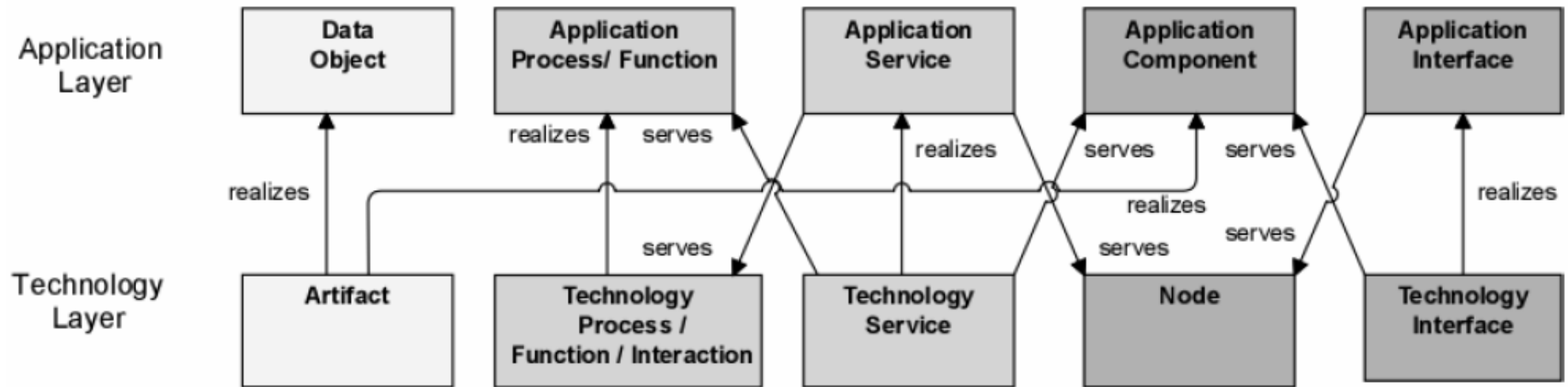
ArchiMate

Přechody mezi vrstvami

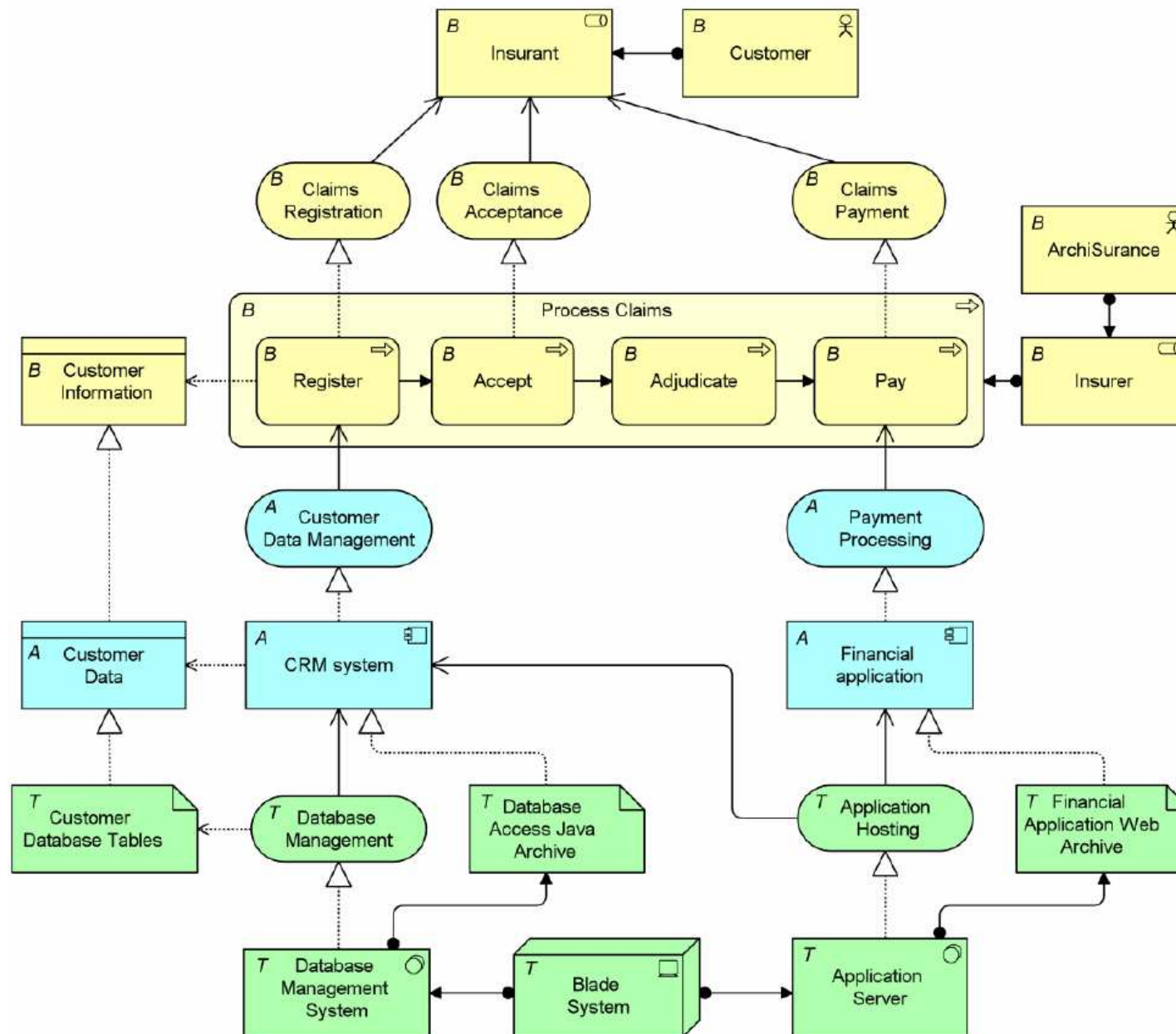
Přechody mezi vrstvami #1



Přechody mezi vrstvami #2



Přechody mezi vrstvami #3



ArchiMate

ViewPoints

View Points

- Viewpoint je konkrétní pohled na část architektury nebo její celek.
- Zpravidla se realizuje formou diagramu. Záleží také na podpoře nástroje (*EA ji má velmi dobrou*).
- Viewpoint se zpravidla zakresluje s ohledem na konkrétního stakeholdera.
- TOGAF definuje taxonomii pohledů.
- Každá organizace si definuje vlastní set pohledů, nicméně existuje několik předdefinovaných základních, které jsou velmi dobře využitelné:

Category: Composition		
Name	Perspective	Scope
Organization	Structure of the enterprise in terms of roles, departments, etc.	Single layer/ Single aspect
Application Platform	Shows structure of a typical application platform and how it relates to supporting technology.	Multiple layer/ Multiple aspect
Information Structure	Shows the structure of the information used in the enterprise.	Multiple layer/ Single aspect
Technology	Infrastructure and platforms underlying the enterprise's information systems in terms of networks, devices, and system software.	Single layer/ Multiple aspect
Layered	Provides overview of architecture(s).	Multiple layer/ Multiple aspect
Physical	Physical environment and how this relates to IT infrastructure.	Multiple layer/ Multiple aspect

Category: Support		
Name	Perspective	Scope
Product	Shows the contents of products.	Multiple layer/ Multiple aspect
Application Usage	Relates applications to their use in, for example, business processes.	Multiple layer/ Multiple aspect
Technology Usage	Shows how technology is used by applications.	Multiple layer/ Multiple aspect
Category: Cooperation		
Name	Perspective	Scope
Business Process Cooperation	Shows the relationships between various business processes.	Multiple layer/ Multiple aspect
Application Cooperation	Shows application components and their mutual relationships.	Multiple layer/ Multiple aspect
Category: Realization		
Name	Perspective	Scope
Service Realization	Shows how services are realized by the requisite behavior.	Multiple layer/ Multiple aspect
Implementation and Deployment	Shows how applications are mapped onto the underlying technology.	Multiple layer/ Multiple aspect

ArchiMate

Odkazy na dokumenty a další studium

Dokumenty pro další studium

Archimate:

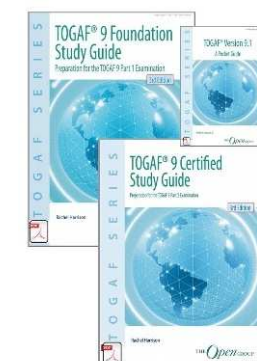
- ArchiMate 3.1 Specification – základní a zásadní publikace, je to vlastní definice jazyka. Nevýhodou je, že některé popisy jsou různě interpretovatelné, pro získání kompletního obrázku o nějakém problému, je potřeba dohledat informace z více kapitol a některé příklady nejsou moc srozumitelné a jsou dost umělé. Při čtení je potřeba si hlídat u elementů z čeho jsou specializované. Lze zakoupit v elektronické i tištěné podobě a lze stáhnout (na omezenou dobu) trial verzi (liší se vodoznakem) a také by měla být po určité době smazána.
- Mastering ArchiMate – výborná kniha, která ukazuje jak používat ArchiMate prakticky. Již existuje pro verzi ArchiMate 3. Lze zakoupit v elektronické podobě.

TOGAF

- TOGAF Version 9.1 – specifikace frameworku v aktuální verzi. Lze zakoupit v elektronické i papírové podobě. Obsahuje nejen specifikaci samotného frameworku, ale i dost užitečných myšlenek a poznatků z oblasti samotné Enterprise a Solution Architektury.
- TOGAF 9.1 Study Guides – průvodce certifikačním programem a zároveň návod, jak efektivně číst vlastní specifikaci TOGAFu

Další doporučená literatura obecně:

- UML 2.4 – UML je základ a vychází z něj i ArchiMate. Must be pro každého IT analytika / architekta.
- BPMN 2.0 – standard pro procesní modelování a prostředek pro rozvinutí business procesů na detailní úrovni.



ArchiMate

Odkazy na zkoušku

Certifikační zkouška

Certifikační zkoušku je nutné složit v certifikovaném zkušebním středisku Open Group Authorized .

Registrace na zkoušku:

1. Nejbližší certifikační středisko je: OKsystem, Na Pankráci 125, Praha 4 (Organizace testů: Simona Dianová tel.: +420 236 072 251 skoleni@oksystem.cz)
2. Certifikační zkoušku nemají uvedenou na stránkách, je nutné kontaktovat OK systém na emailu uvedeném výše a objednat termín.
3. Úroveň znalostí: ArchiMate® 3 Foundation
4. Název zkoušky: ArchiMate 3 Part 1
5. Typ zkoušky: prostřednictvím www aplikace.
6. Jazyk zkoušky: Angličtina
7. Otázky: 40 otázek během 60 minut (může být poskytnut dodatečný čas pro kandidáty, pro něž není angličtina primární jazyk).
8. Typ odpovědí: výběr z možností.
9. Úspěšné složení: 60% z otázek.
10. Pomůcky: bez pomůcek
11. Další pravidla: zákaz komunikace s ostatními v místnosti, personál nesmí odpovídat na otázky ohledně obsahu zkoušky. Kandidát nesmí mít v místnosti nic, kromě naléhavých lékařských pomůcek (ani vodu například), žádnou tašku, peněženku, mobil, vše se odkládá do zamčené skříňky.
12. Dohled: stálý kamerový dohled.
13. Vyhodnocení testu: okamžité, certifikát je vystaven společností Open Group elektronicky (po následném dokončení certifikace kandidátem již mimo OK systém – <http://www.opengroup.org/certifications/how-to-complete>).
14. Dva doklady totožnosti s sebou (s fotografií).
15. Je nutné mít předem vytvořen účet (pokud ho nemáte) na stránkách: <http://www.opengroup.org> (login / create account).

U testu přemýšlet a pročítat otázky tam a zpět – dost často se stane, že jedna otázka odpoví na druhou – například jeden symbol je zobrazen s otázkou na jeho název a druhá otázka se ptá do jakého aspektu tato Business Function patří...a stačí vám zodpovědět správně 24 otázek.

ArchiMate

Test - elektronicky

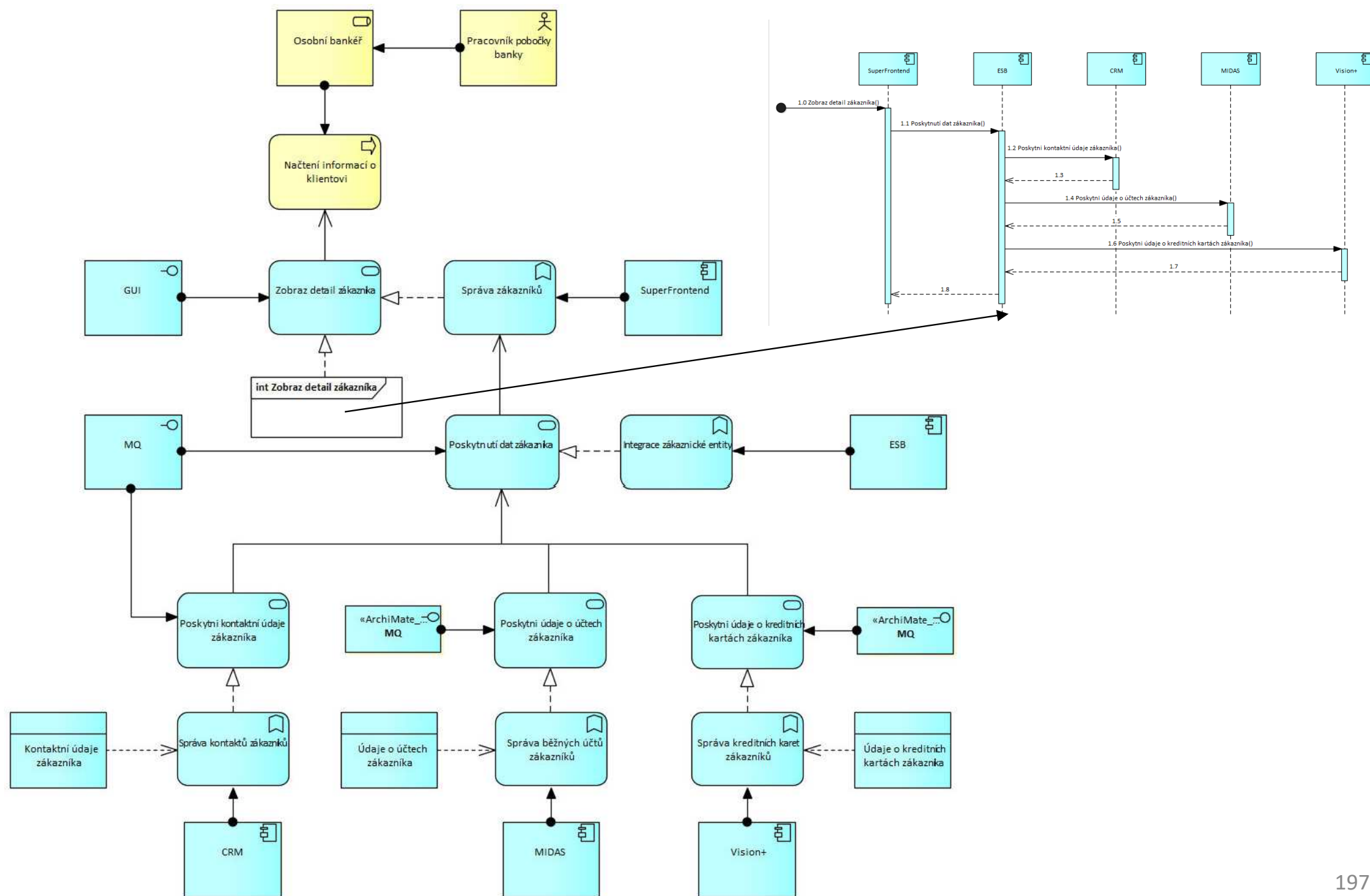
ArchiMate

Vzorová řešení (patterns)

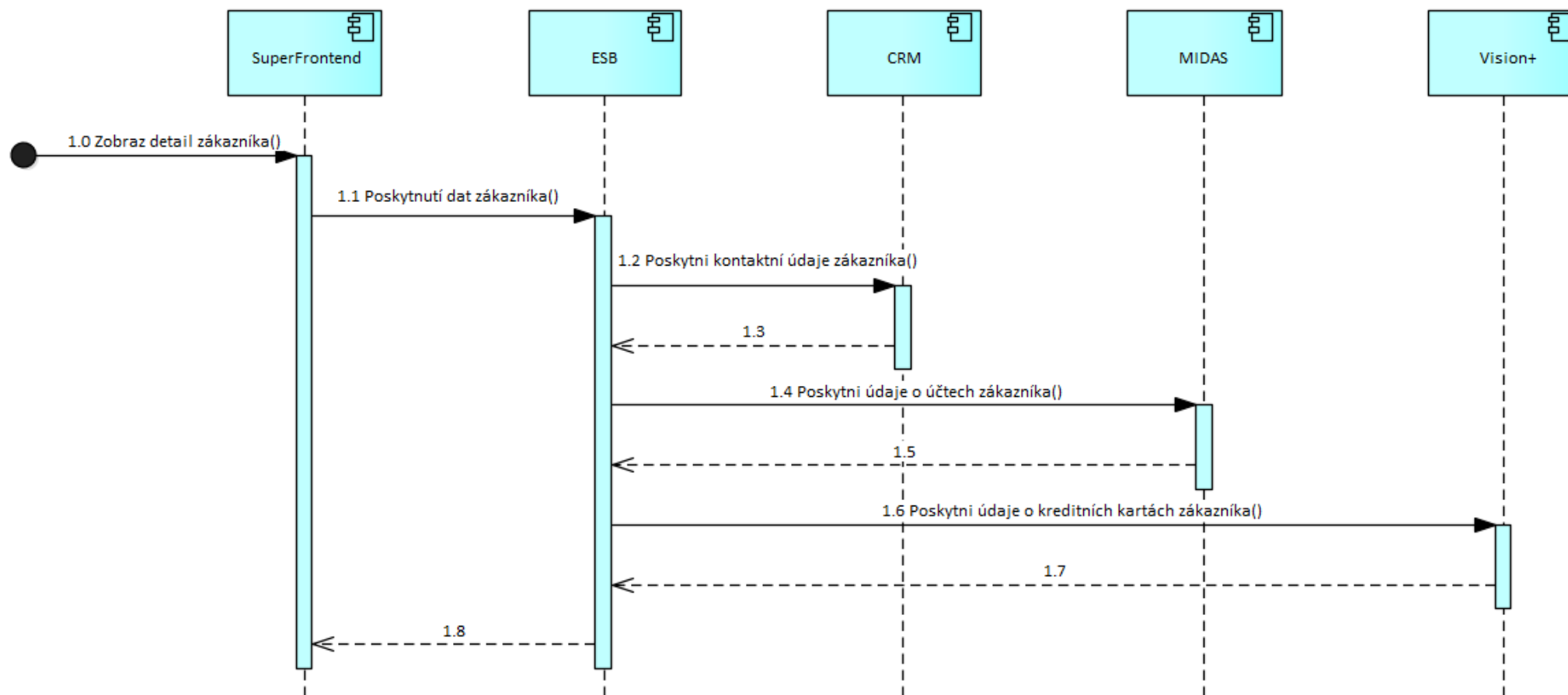
ArchiMate

Integrační patterny

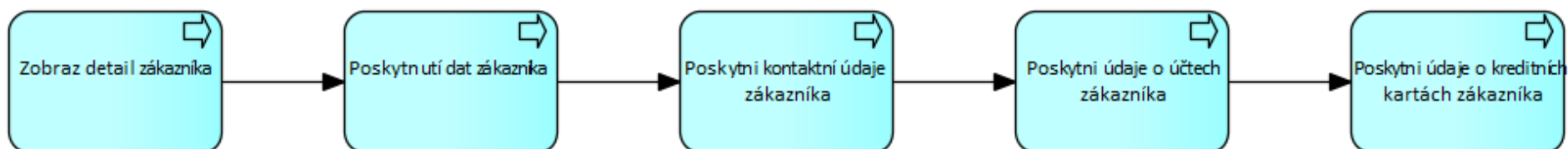
Zakreslení integrace pomocí služeb + vazby Serving + Sekvenčního diagramu #1



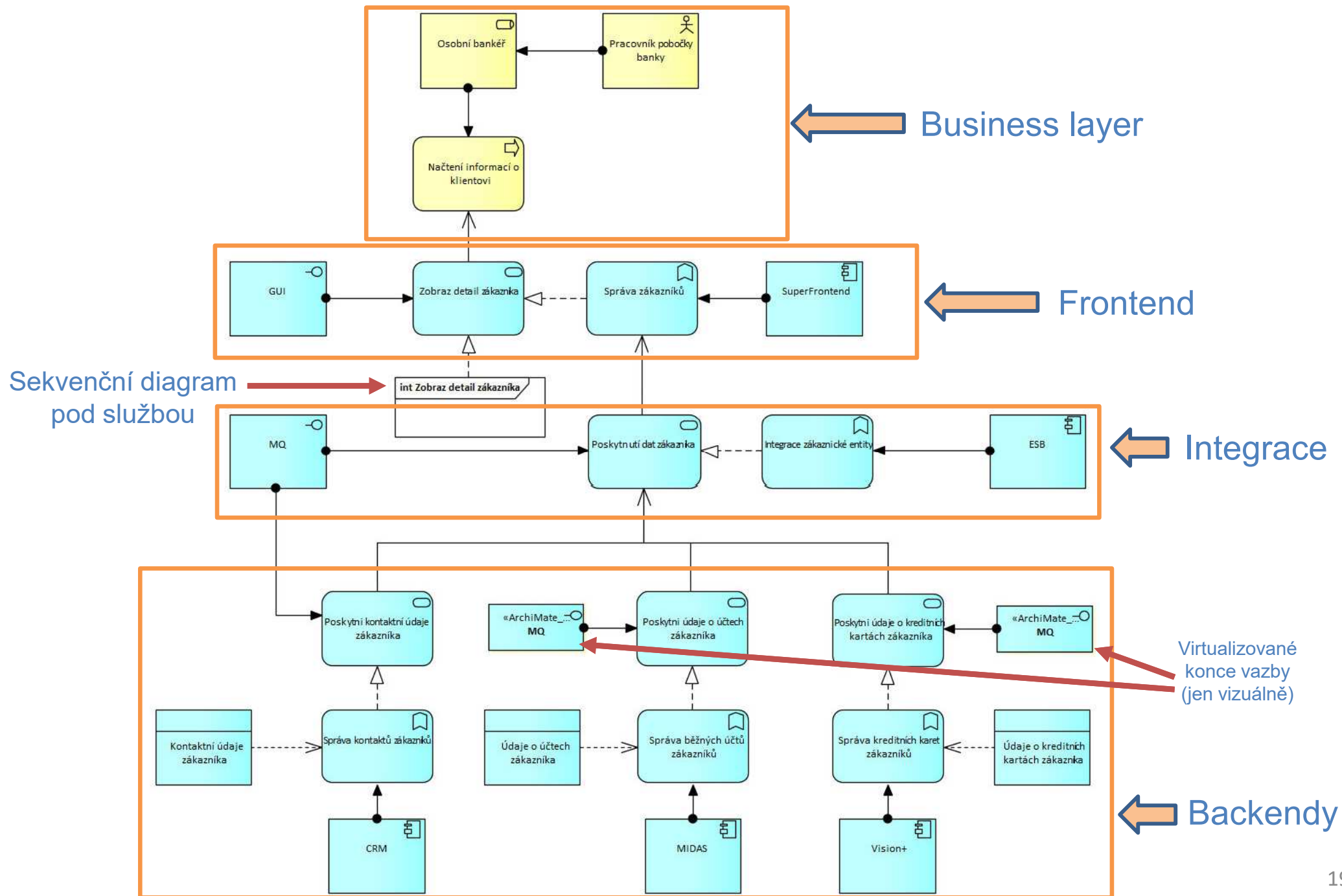
Zakreslení integrace pomocí služeb + vazby Serving + Sekvenčního diagramu #2



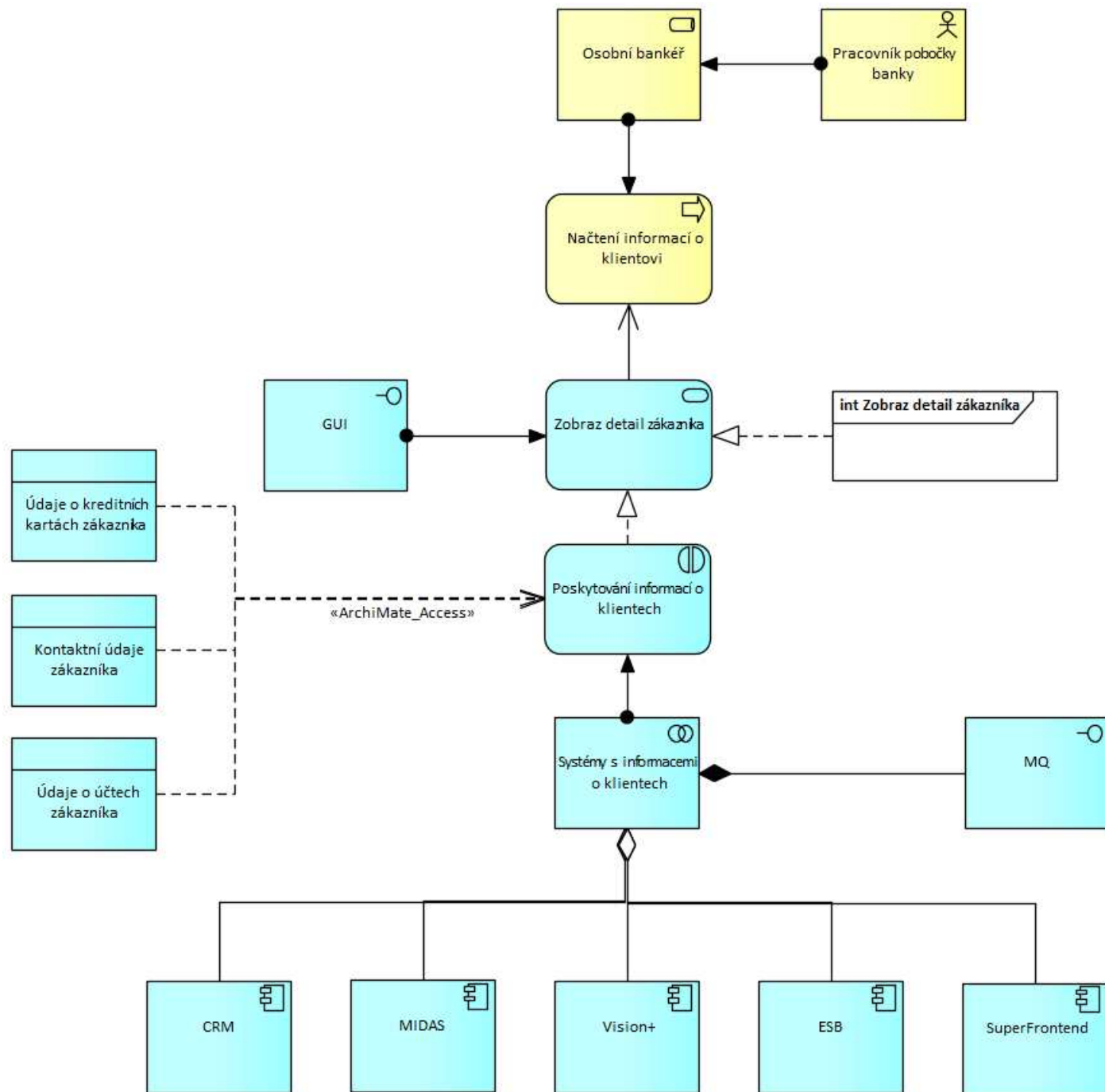
Sekvenční diagram může být nahrazen aplikačním procesem



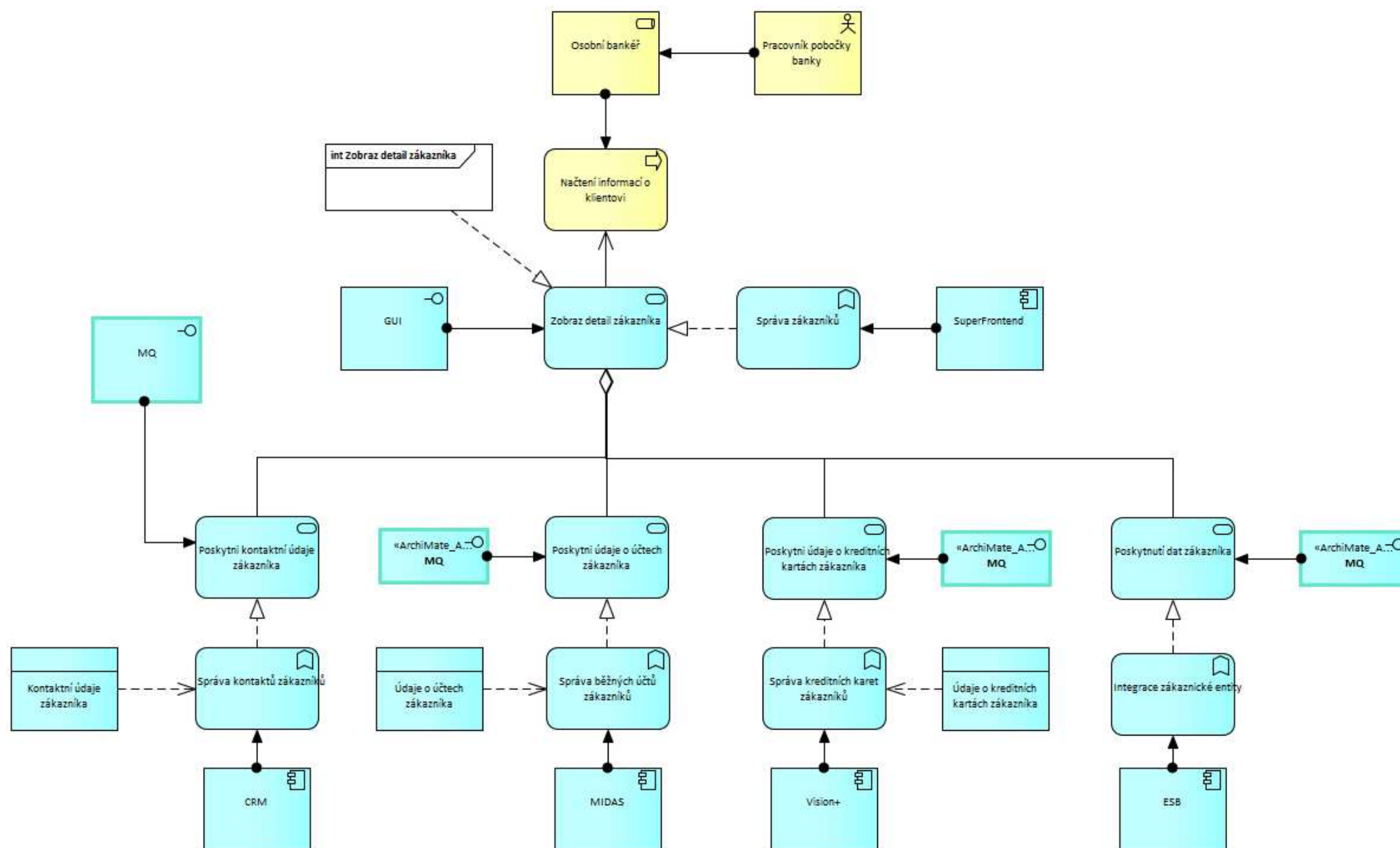
Zakreslení integrace pomocí služeb + vazby Serving + Sekvenčního diagramu #3



Zakreslení integrace pomocí aplikační kolaborace #1



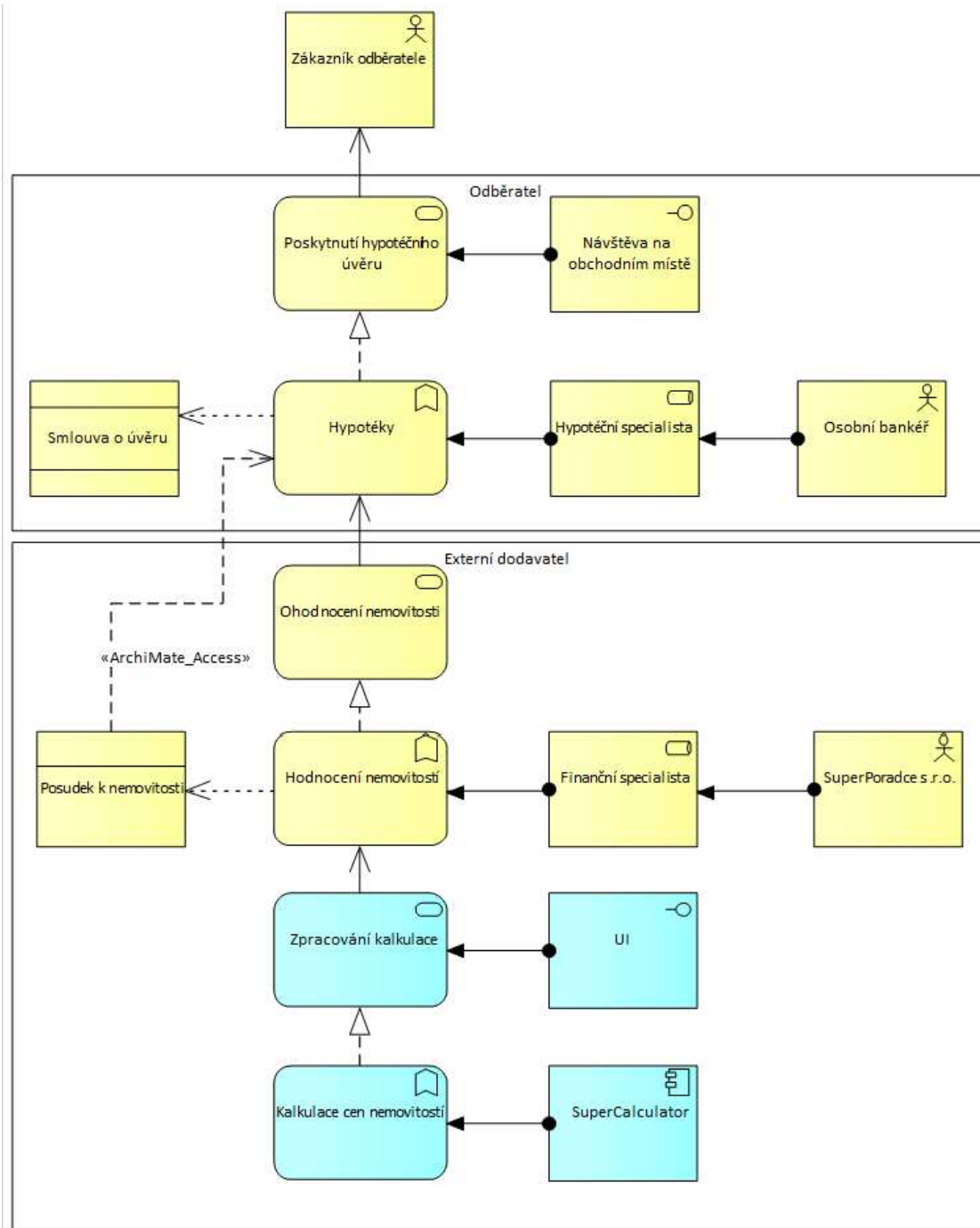
Zakreslení integrace pomocí agregace služeb #1



ArchiMate

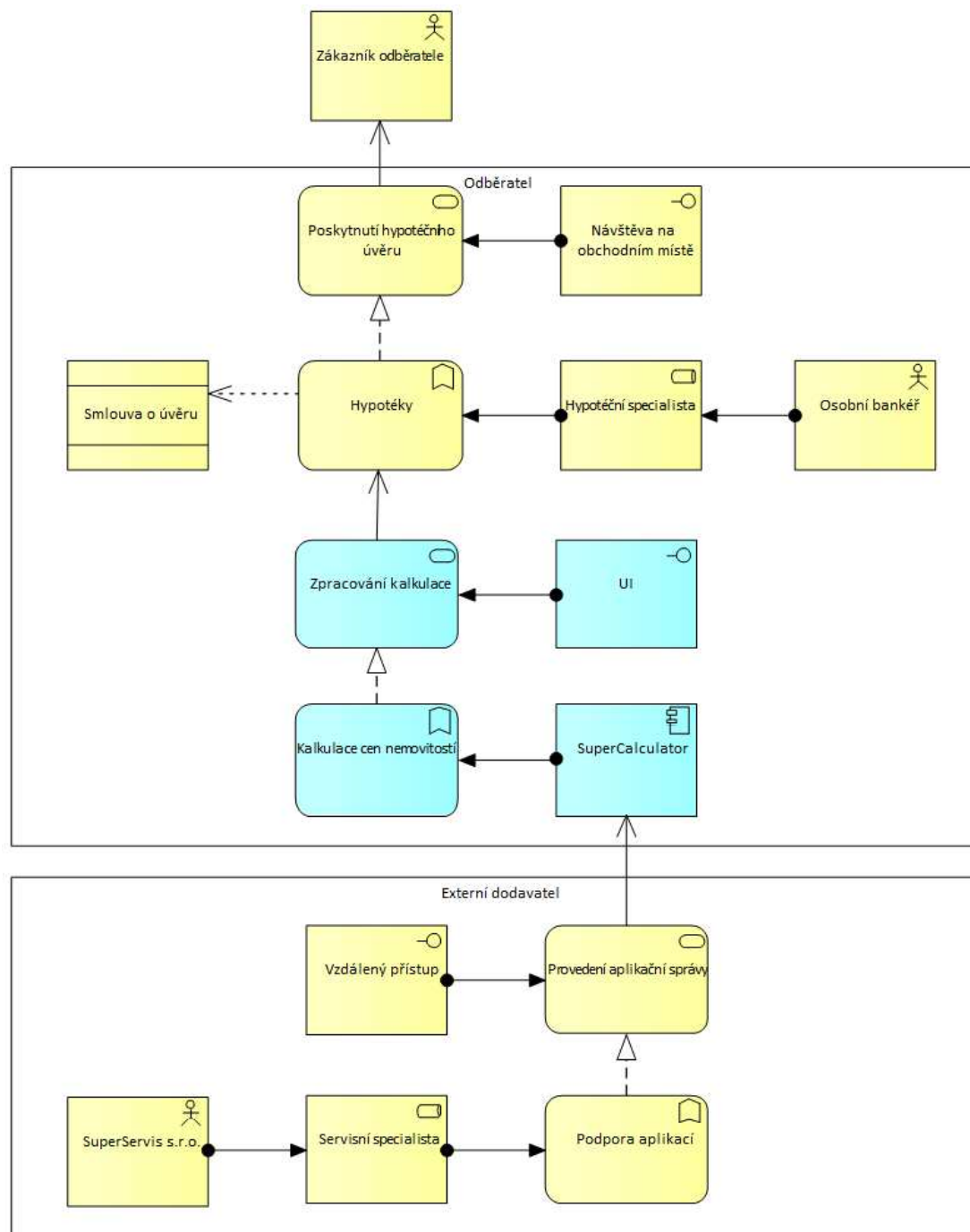
Modelování dodavatelů

Dodavatel podporuje svou aplikací business process / funkci odběratele



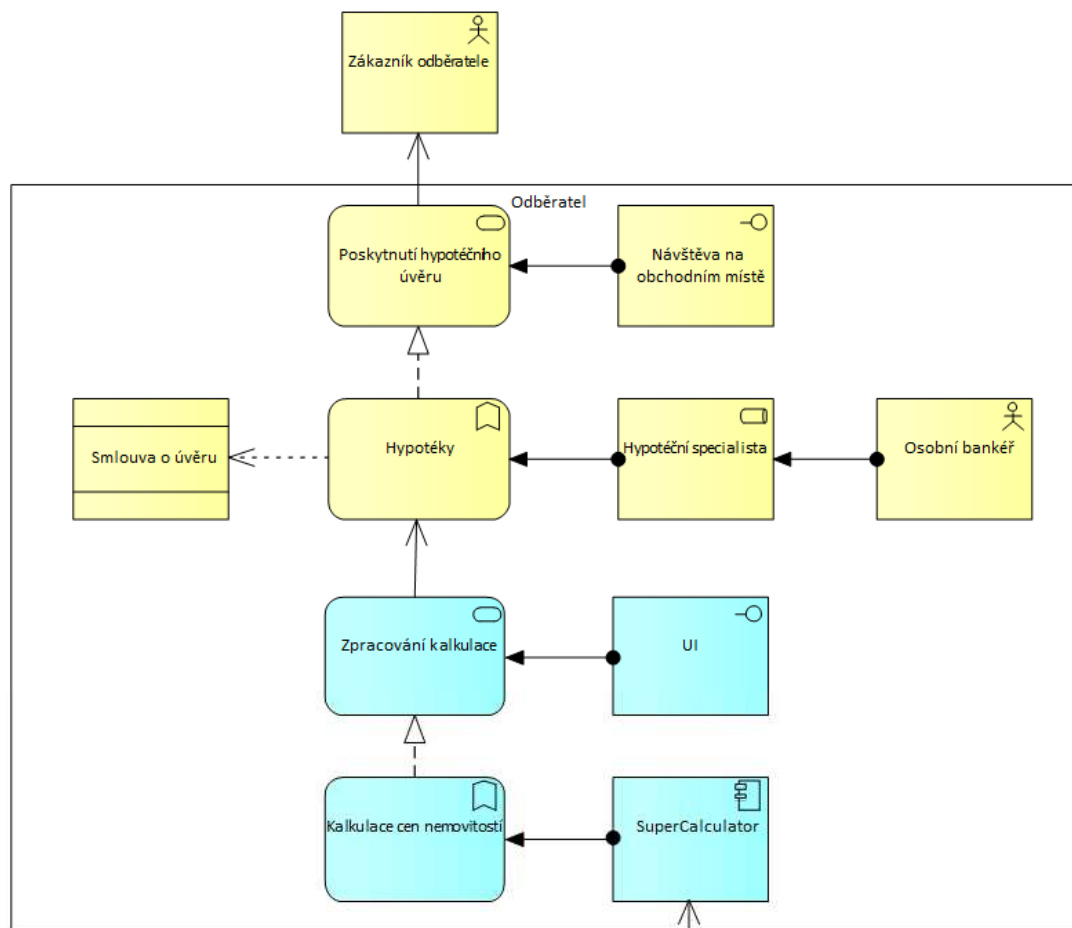
Dodavatel podporuje svou činností business odběratele a využívá k tomu svou vlastní aplikaci.

Dodavatel podporuje aplikaci odběratele

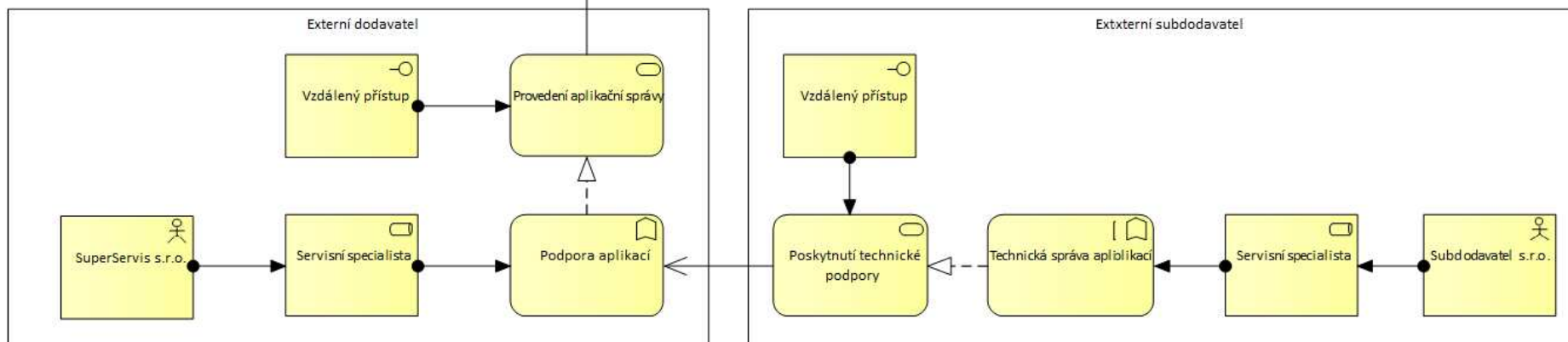


Aplikace patří odběrateli a je podporována externím dodavatelem.

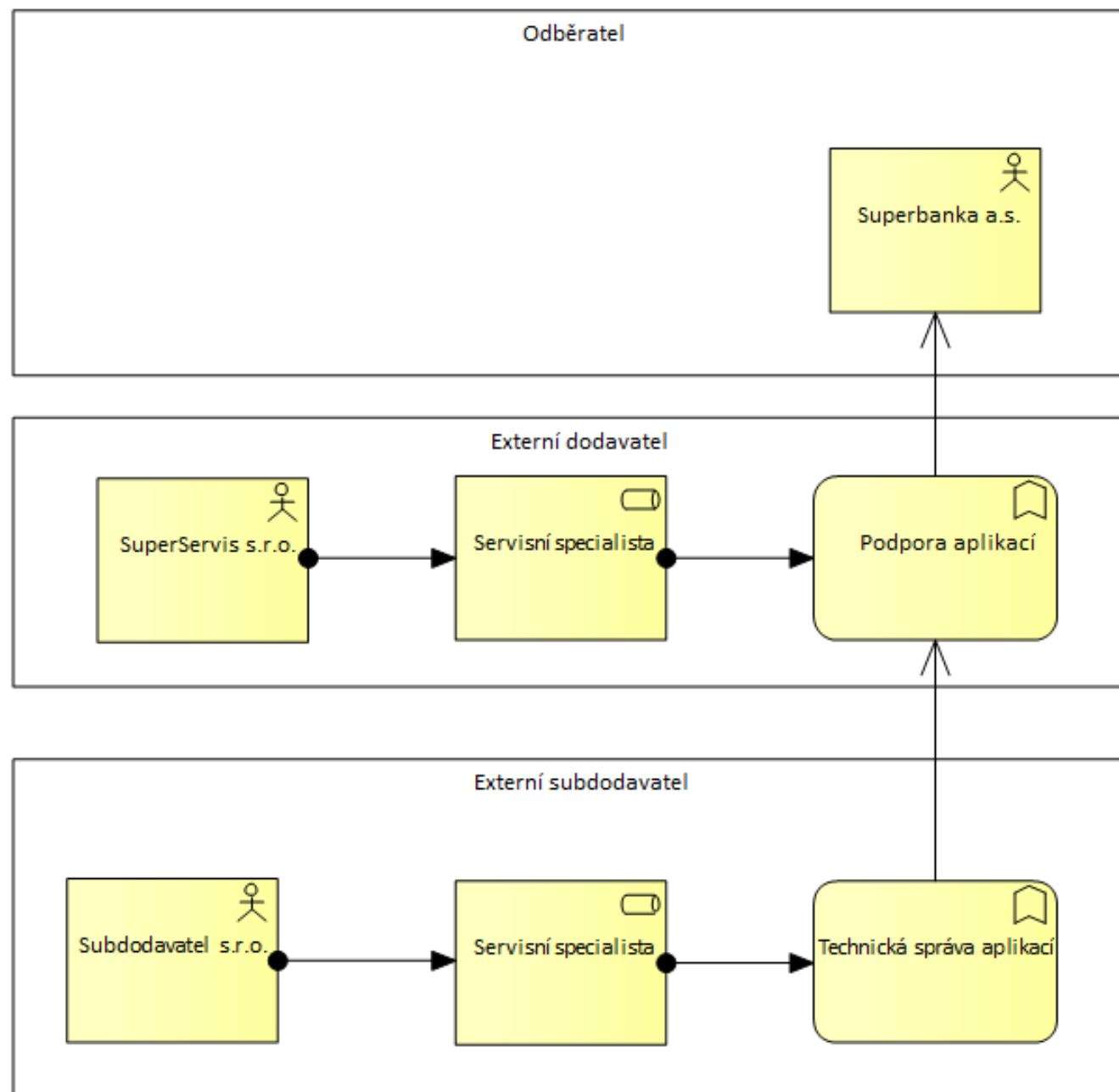
Dodavatel podporuje aplikaci odběratele a má subdodavatele



Aplikace patří odběrateli a je podporována externím dodavatelem, který navíc využívá subdodavatele.



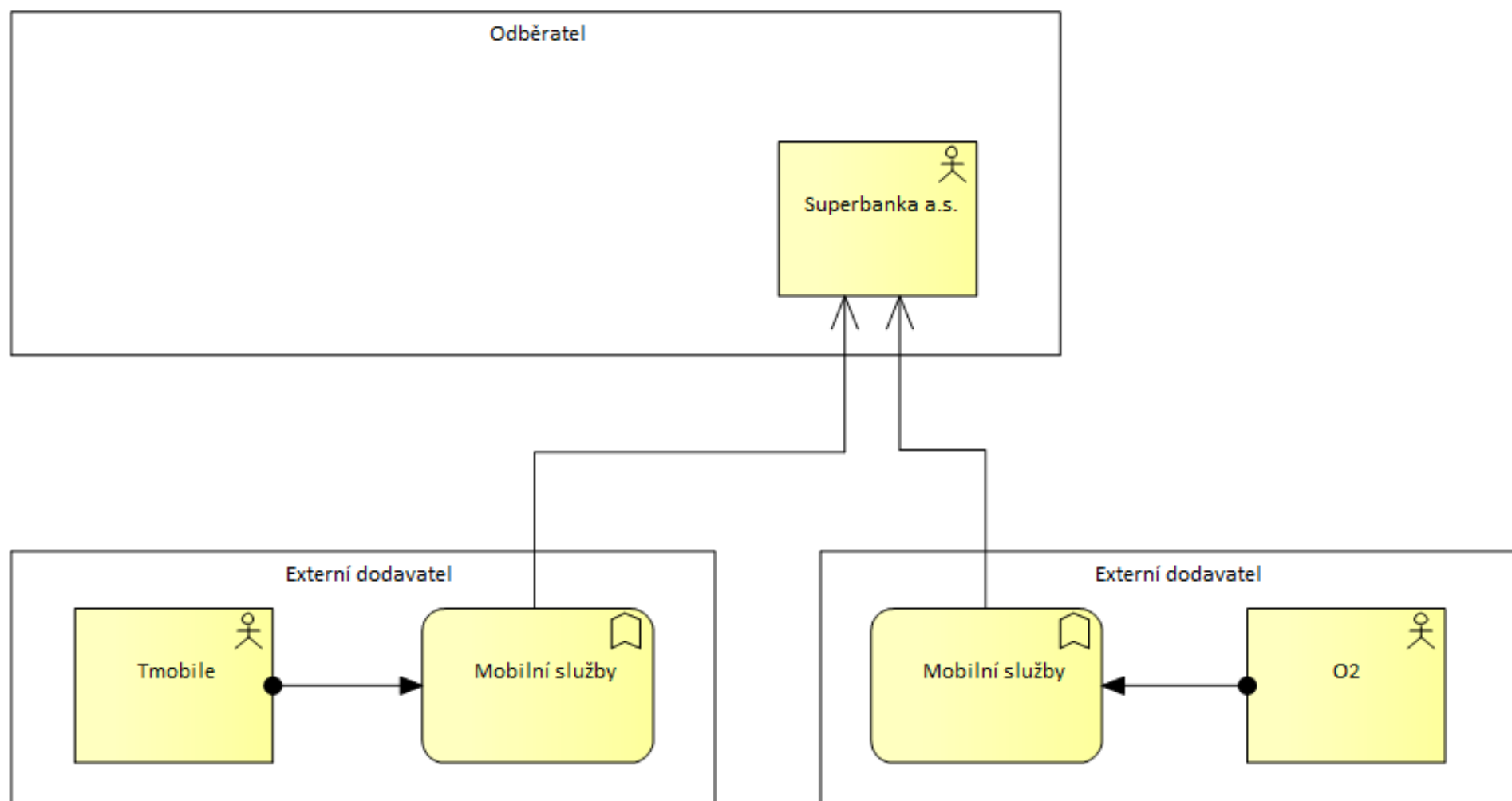
Dodavatelský řetězec zjednodušeně



Tento diagram znázorňuje to, že externí dodavatel dodává odběrateli služby – je zobrazeno které, ale ne, k čemu je odběratel využívá. Z tohoto důvodu je i vypuštěna úroveň služeb. Externí dodavatel má dále subdodavatele.

Dva dodavatelé dodávají stejnou službu

Dva dodavatelé dodávají stejnou službu odběrateli (například za jinou cenu).



Otázky & Diskuze

