Avancier

Slide show of the Paper
"Architecture Harmonization
Harmonizing TOGAF with other approaches"

This document updates and extends research done by Avancier Ltd for the British Computer Society into harmonization of different enterprise architecture standards and sources.

Aims

- This slide show presents TOGAF in a coherent and consistent way.
- And relates it to other architecture standards and sources

  - Part 1: Harmonising TOGAF with itself
  - Part 2: Harmonising TOGAF with data architecture
  - Part 3: Harmonising TOGAF with SOA
  - Part 4: Harmonising TOGAF with Zachman
  - Part 5: Harmonising ArchiMate with TOGAF
  - Part 6: Harmonising TOGAF with ArchiMate
  - Part 7: Radically changing TOGAF to fit ArchiMate?
  - Part 8: Harmonising TOGAF with Business Capabilities
  - Part 9: Harmonising TOGAF with Value Streams
  - Part 10: Adapting TOGAF to use BA terminology

- An aim is to help people ensure piecemeal change requests do not undermine the coherent and consistent view presented herein.

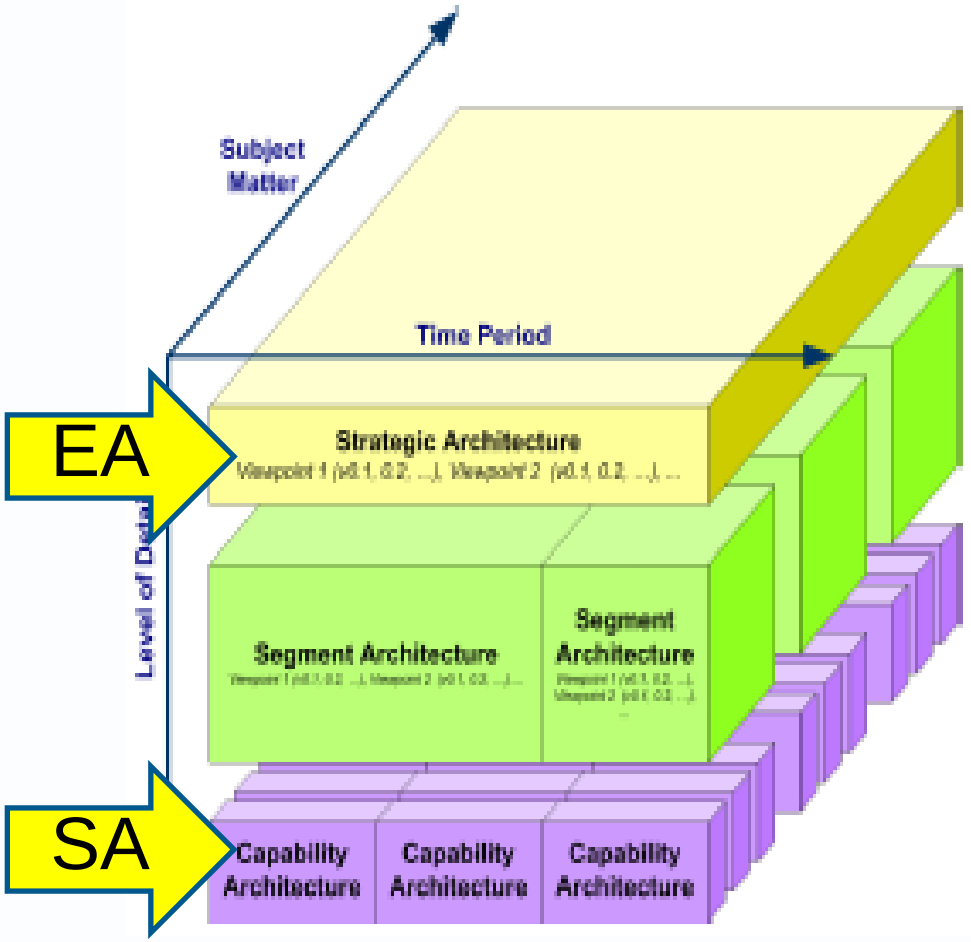
PART 1: Harmonising TOGAF with itself

- TOGAF sets out a method (ADM) with a cycle of 8 phases.
- It develops an architecture in response to a request for work
- The architecture develops from required services, to architecture building blocks to solution building blocks.

ADM Phase	Deliverable (v1 on)	Services & Building Blocks	Enterprise Continuum	Enterprise Repository
Phase A onwards	Architecture Req'ments Specification	Business & Application Service Contracts	Requirements & Context	Requirements Repository
Phases B, C, D onwards	Architecture Definition Document	Architecture Building Blocks	Architecture Continuum	Architecture Repository
Phases E and F	Architecture Road Map	Solution Building Blocks	Solutions Continuum	Solutions Repository
Phases G and H	Architecture Change Requests		Deployed Solutions	

Two very different levels of architecture

- The ADM is flexible, highly iterative, and applied to different scopes from broad to narrow
- It is important to understand the difference between
  - broad enterprise architecture
  - narrow solution architecture.



Different levels of architecture – different artefacts

Avancier

Enterprise / Strategy / Portfolio level

Primarily for use in analysis and direction setting at a strategy or portfolio level.

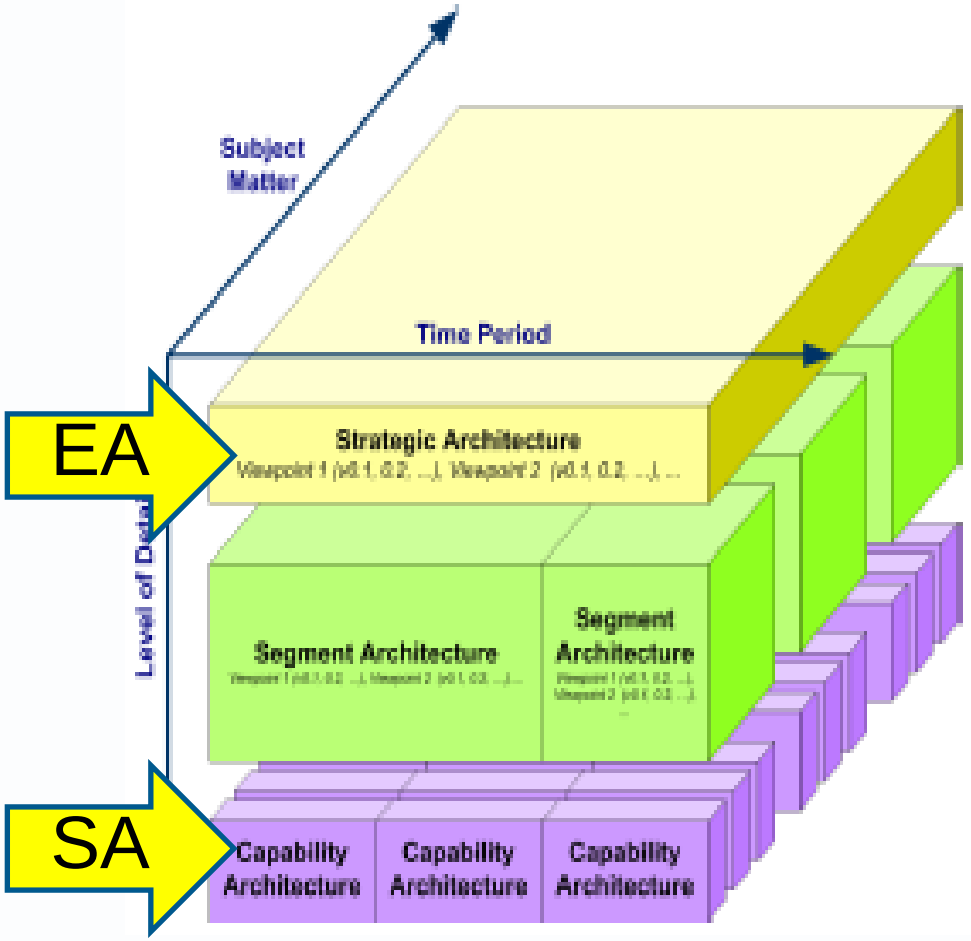
Used for gap analysis, cluster analysis, impact analysis and traceability analysis.

Mostly catalogs and matrices (organization and function decomposition diagrams can be seen as hierarchical catalogs.)

Solution or Capability Increment level

Primarily for use during an ADM cycle at the solution or capability increment level.

Mostly diagrams.



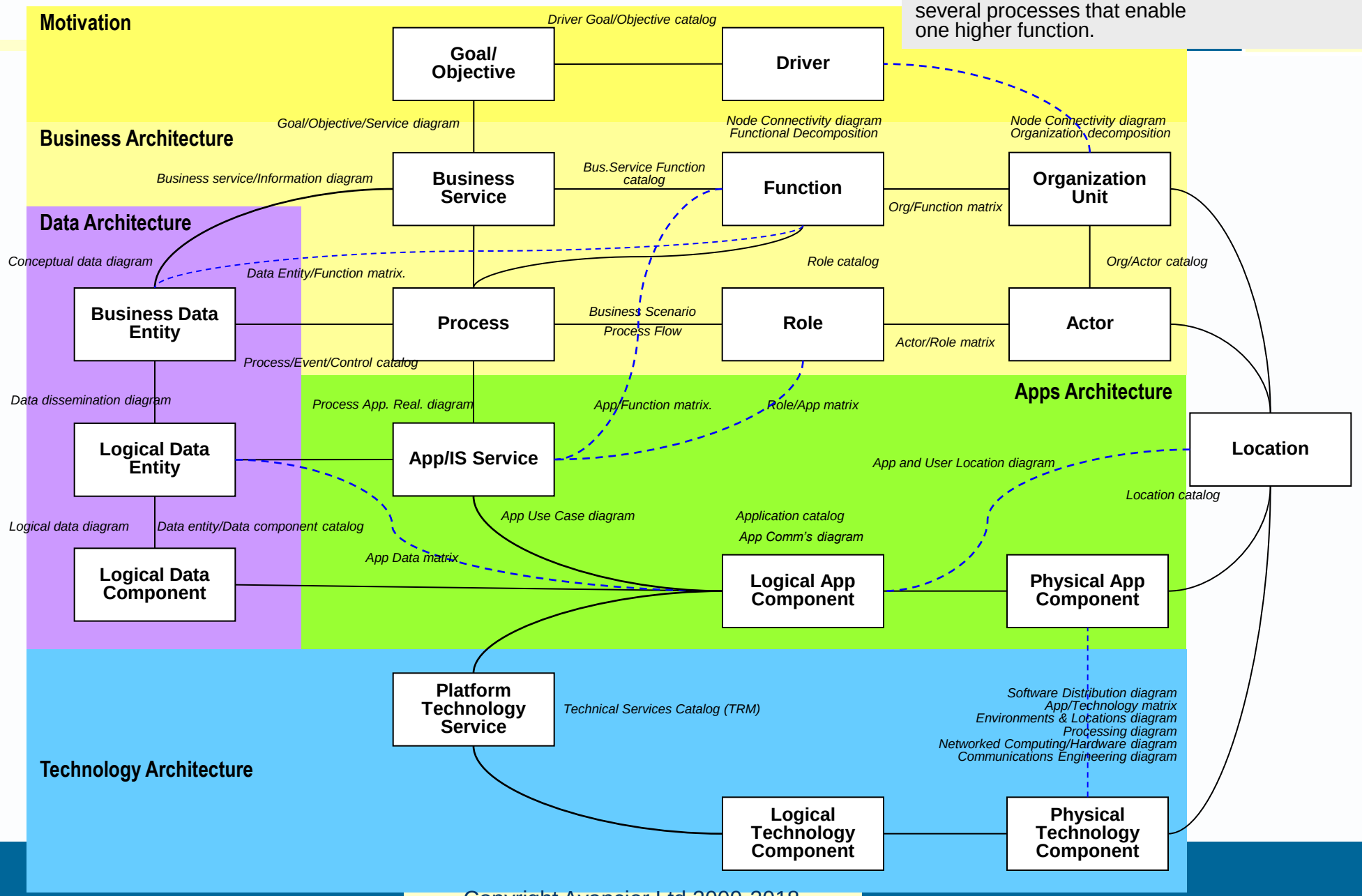
Different levels of architecture – different artefacts

Avancier

	Enterprise / Strategy / Portfolio level artefacts	Solution or Capability Increment level artefacts
Motivation	Driver Goal/Objective Catalog	Goal/Objective/Service Diagram
Business	Organization Decomposition Diagram Node Connectivity Diagram (physical or logical) Functional Decomposition Diagram Function/Org Matrix Role Catalog Business Function/Service Catalog Process/Event/Control/Product Catalog	Process Flow Diagram Business Scenario Actor/Role Matrix Organization/Actor Diagram
Applications	Application Portfolio Catalog Application/Function Matrix Role/Application Matrix Application Communications Diagram	Process Application Realization Diagram Application Use Case Diagram Application User Location Diagram Software Engineering Diagram Software Distribution Diagram
Data	Conceptual Data Diagram Data Entity/Function Matrix Application/Data Matrix Data Entity/Data Component Catalog Data Dissemination Diagram	Business Service/Info Diagram Logical Data Diagram Data Security Diagram Data Lifecycle Diagram Data Migration Diagram
Technology	Technology Standards Catalog Technology Portfolio Catalog Technology Services Catalog (TRM) Technology/Application Matrix	Environments and Locations Diagram Processing Diagram Networked Computing/Hardware Diagram Communications Engineering Diagram Platform Decomposition Diagram

A meta model abstracted from c40 artefacts

All entities may be recursively composed and decomposed, and all relationships are many to many. So for example, a process may coordinate several low level functions, and be one of several processes that enable one higher function.



Notes

- 1: This meta model shows the analysis necessary to ensure the coherence and consistency of TOGAF. Simplifying and clarifying the artefact definitions (as in the slide show 1a) would be a major improvement.

- 2: Some artefacts in the bottom right corner are products of architecture implementation (during ADM phase G) rather than of enterprise architects.

- 3: Most entities may be recursively composed and decomposed, and all relationships are many to many. So for example, a process may coordinate several low level functions, and be one of several processes that enable one higher function.

- 4: The TOGAF meta model is not a picture for untrained people, or a CASE tool database schema. The meta model draws a distinction that is not clear in the artefact definitions - **the distinction between logical and physical components, discussed below.**

TOGAF's general principles

- Enterprise architects separate logical specifications from physical realizations.

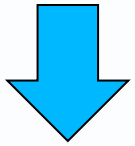
- Distinguish
  - *external* views (services) from *internal* active structures (components).
  - *logical* structures and components from *physical* ones

  - Logical:
    - "An implementation-independent [portable, vendor-neutral] definition of the architecture."

  - Physical:
    - "A description of a real-world entity" that is still "considerably abstracted from implementation".

TOGAF's general principles

- Enterprise architects should separate logical specifications from physical realizations.



Motivations and constraints	Passive structures	Behaviors	Logical structures	Physical structures
	Things that are acted on	Things that happen	Specifications of thing that act	Things that act
Drivers Goals/Objectives	Business Data Entities	Business Services Processes	Functions, Roles	Organization Units, Actors
	Logical Data Entities	IS Services	Logical Application Components	Physical Application Components
		Platform Services	Logical Technology Components	Physical Technology Components

Logical specification tools

- Enterprise architects are concerned with describable and testable business systems.

- A system is described as a collection of
  - interrelated and repeatable activities (aka behaviors)
  - performed by actors (aka active structures).

- System theory gives architects tools for specifying systems that are logical and independent of computing.

Logical specification tools

Avancier

	Viewpoint	Logical description format
Behavior	External view	Service contract — Business Service; IS Service; Technology Service
	Internal control logic	Process flow diagram — Process
Active structure	External view	Service portfolio or interface — Function; Role; Logical App Component; Logical Technology Component
	Internal connectivity	Node connectivity diagram (communication between active structures)
Passive structure	Memory / data store	Entity-attribute-relationship model — Business Data Entity; Logical Data Entity; Logical Data Component
	Message / data flow	Regular expression structure (data flow structures)

Note: an entity-attribute-relationship model

- First and foremost, defines an ontology...
  - “concepts and categories in a subject area that shows their properties and the relations between them.”
- It defines a domain-specific vocabulary used in communicating facts about things of interest to a business.
- It is therefore helpful in defining the contents of
  - messages (data flows) and
  - memories (data stores).

TOGAF promotes abstraction

- The separation of
  - logical “architecture buildings blocks” from
  - physical “solution building blocks.”

Architecture Continuum **Architecture Repository**	**Architecture Building Blocks**	implementation-independent logical component specifications - which are "realised by” SBBs
Solutions Continuum **Solutions Repository**	**Solution Building Blocks**	nominate/describe physical components to be hired, bought or built (still "considerably abstracted from implementation“)

- runs all the way through TOGAF’s
  - ADM,
  - deliverables,
  - meta model,
  - enterprise continuum
  - enterprise repository.

TOGAF specification levels

Avancier

Enterprise Continuum Enterprise Repository	Level of specification	
	1 EA/Strategic Architecture & Business Planning	gives rise to 2
Requirements & Context **Requirements Repository**	**2 Architecture Requirements & Service Contracts**	which are clustered and assigned to 3
Architecture Continuum **Architecture Repository**	**3 Architecture Building Blocks**	implementation-independent logical component specifications - which are "realised by" 4
Solutions Continuum **Solutions Repository**	**4 Solution Building Blocks**	nominate/describe physical components to be hired, bought or built. still "considerably abstracted from implementation" and truly realised by 5
Deployed Solutions	**5 Deployed Solutions**	passive configuration structures
		active (run-time) structures and behaviors

TOGAF promotes abstraction

- TOGAF presumes enterprise architects work at levels 1, 2 and 3, well above the artefacts deployed at level 5.

Enterprise Continuum	Enterprise Repository	Level of specification
		1 EA/Strategic Architecture & Business Planning
Requirements & Context	Requirements Repository	2 Architecture Requirements & Service Contracts
Architecture Continuum	Architecture Repository	3 Architecture Building Blocks
Solutions Continuum	Solutions Repository	4 Solution Building Blocks
Deployed Solutions		5 Deployed Solutions

- It cannot dictate where each specification level is stored, but presumes each is traceable to the one above and below.
- You can tag each building block's position in other dimensions, such as from generic-to-specific (Foundation to Organization).

Mapping abstraction levels to architecture domains

Avancier

ADM Deliverable	Enterprise Continuum Enterprise Repository	Services & Building Blocks	Business domain entities	Applications domain entities	Data domain entities	Technology domain entities
EA/Strategic Architecture			Function and organization hiearchies	Application portfolio catalog	Business data entity catalog	Technology services catalog (TRM)
Architecture Req'ments Specification	Requirements & Context Requirements Repository	Business & Application Service Contracts	Business Services, Processes	App/IS Services		
Architecture Definition Document	Architecture Continuum Architecture Repository	Architecture Building Blocks	Functions, Roles	Logical App Components	Logical Data Entities & Data Components	Logical Technology Components
Architecture Road Map	Solutions Continuum Solutions Repository	Solution Building Blocks	Organization Units, Actors	Physical App Components	Physical Data Components	Physical Technology Components
Architecture Change Requests	Deployed Solutions		Identity Management	IT Configuration Management (CMDB)		
			Business & IT Operations			

Note 1: Architecture work outside of an ADM cycle

- Strategic level catalogs and matrices may be documented and maintained before and outside of an ADM cycle.

- E.g. TOGAF suggests using functional decomposition for heat mapping and scoping changes planned via an ADM cycle.

Note 2: Early selection of SBBs

- You may define technology solutions while working on the products of phase A to D. Phase A might start with a request to implement a particular application or technology in one organization unit.

- That does not mean documenting that solution in the requirements repository or architecture repository. It means starting phase E early and populating the solutions repository with the (likely) solution building block.

- The TOGAF principle is to define that solution also in a logical (implementation-independent, portable, vendor-neutral) architecture definition. That (rather than solution documentation) is what architects maintain in phase H: “architecture change management”.

Part 2: Harmonising TOGAF with Data Architecture

- Data architecture is about
  - memories,
  - messages and their
  - meanings.

- It is about
  - data in stores (at rest),
  - data in motion (in flows) and
  - the qualities of the data in the stores and flows.

Data at rest – data stores

- *Business Data Entity*
  - A thing or event of interest that the business must remember to complete its processes.
  - It could be a normalised entity or a document; it could be recorded in several data stores.
  - The conceptual data model serves as catalog of business data entities important to business operations, especially ones that are stored in several data stores/components.
- *Logical Data Entity*
  - The appearance of a business data entity in one data store/component.
- *Logical Data Component*
  - The implementation-independent documentation of the data encapsulated in a data store/component.
  - A logical data model is usually an entity-attribute-relationship model, which should support known update and enquiry access paths.

Physical data components?

- This is tool-specific documentation of the data encapsulated in a data store/component.
- The data structure is expressed in the form required by particular data storage technology (CODASYL, RDMS, Document Store, XML schema, etc).

- Physical data components are not really needed in the TOGAF meta model.
- Enterprise architects don't document database schema, deployed artefacts (as in a CMDB), or storage area networks.
- They can instead model the physical application component that encapsulates a physical data component.

Data in motion – data flows

- Why don't I/O data flows appear in the data architecture section of the TOGAF meta model?
- Probably because I/O data flows are attributes of service contracts - addressed in the next SOA extension.

- Does TOGAF expect *all* I/O data flows to be documented in service contracts?
- Then it must extend the concept of a service to include ones that are pushed as well as pulled/requested.
- The trigger for the provision of such a service may be a time event, or an internal state change event.

Part 3: Harmonising TOGAF with SOA

- The Open Group's vision is of the "Boundaryless Information Flow", and open, vendor-neutral specification of systems.

- To this end, TOGAF promotes specification of systems, and components within them, by the services they offer.

The typical contents of a service contract

Avancier

Name	Capture order
Goal	As implied by the name above and exit facts below
Roles	Customer Sales person
Entry criteria	Trigger: Visit customer at scheduled time Input: Customer details Preconditions: Sales visit agreed and scheduled
Exit criteria	Outputs or products: Signed order Post conditions: Order captured and recorded
Non-functional qualities	Duration: 1 hour Throughput: 2 per day per salesman Availability: Working hours

Note measures are associated with service qualities and results

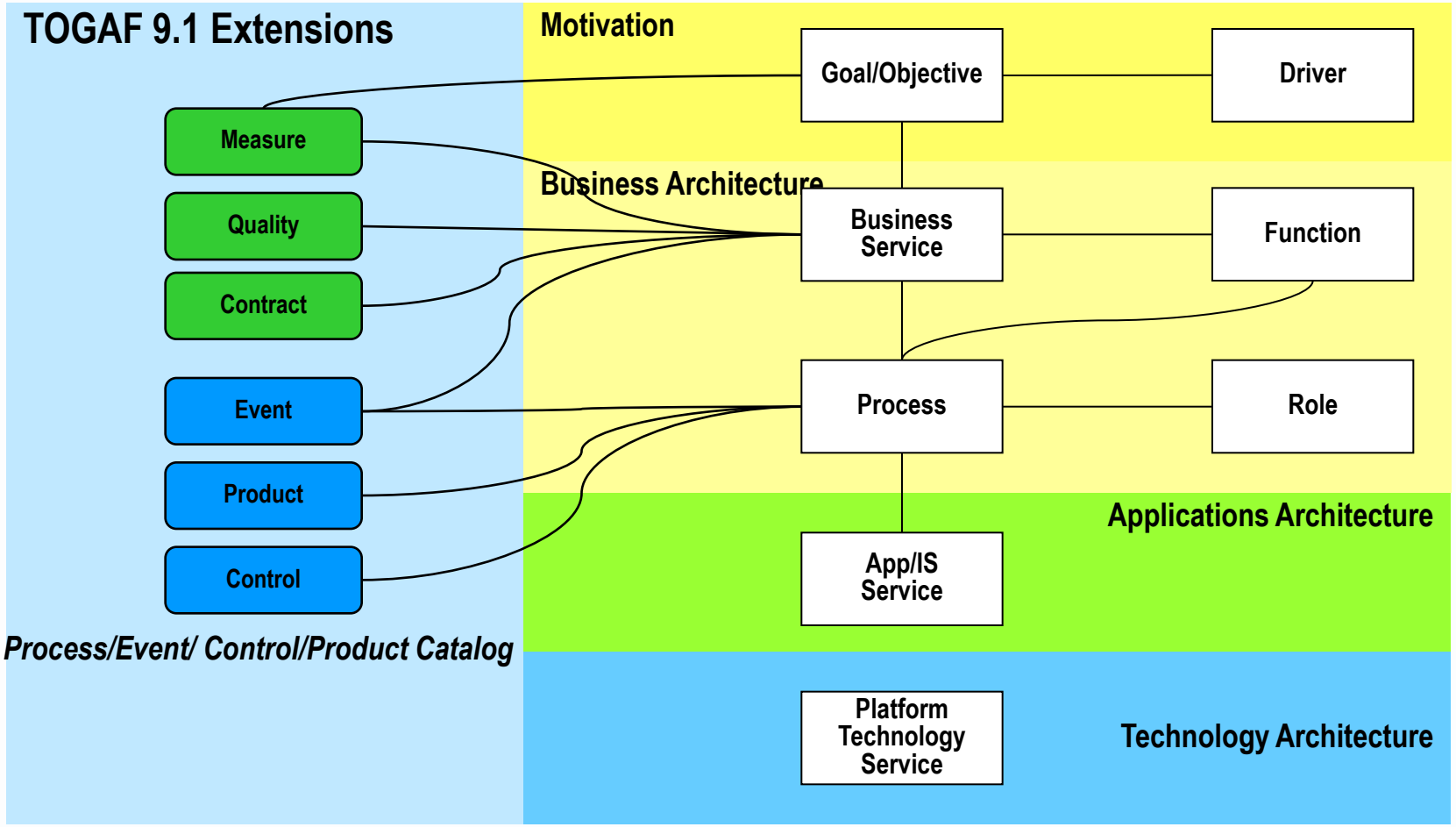
Architects record desired measures, which guide and constrain system design.

Managers may record actual measures of the performance/outputs/effects of system operation.

TOGAF 9.1 meta model extensions

Avancier

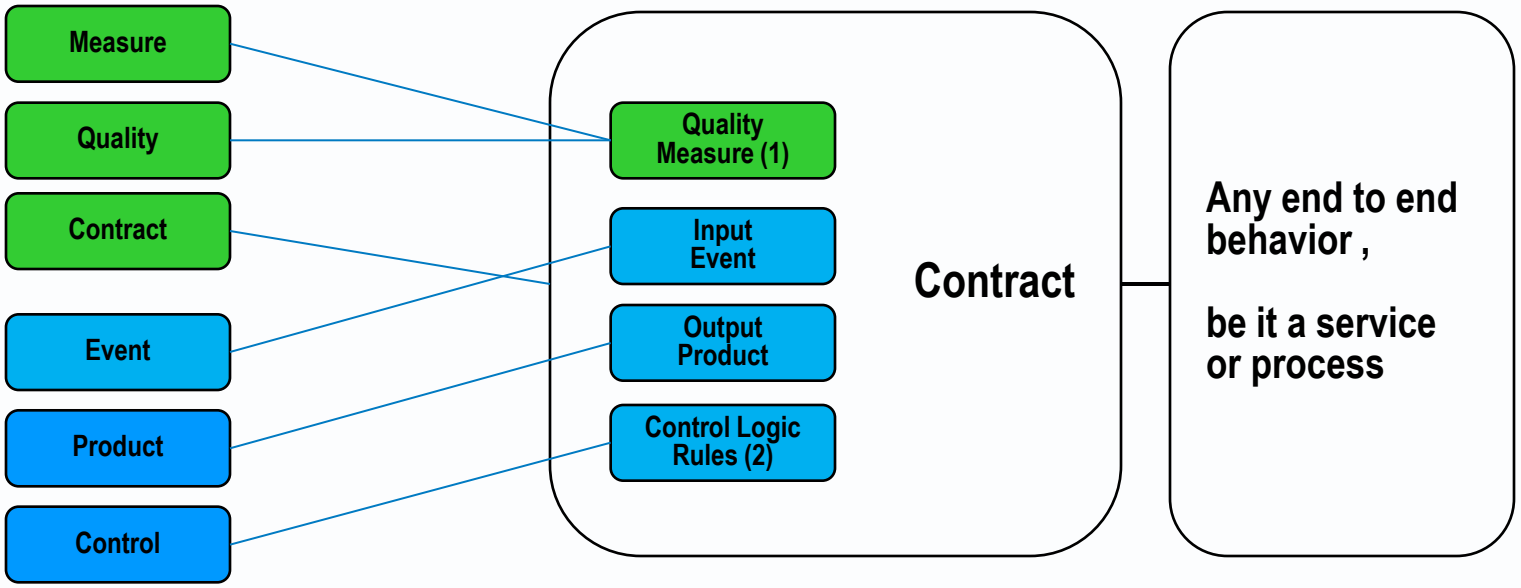
- Already include attributes relating to motivation and service definition



Adapting those extensions to form a simple behavior contract

Avancier

TOGAF 9.1 extensions consolidated

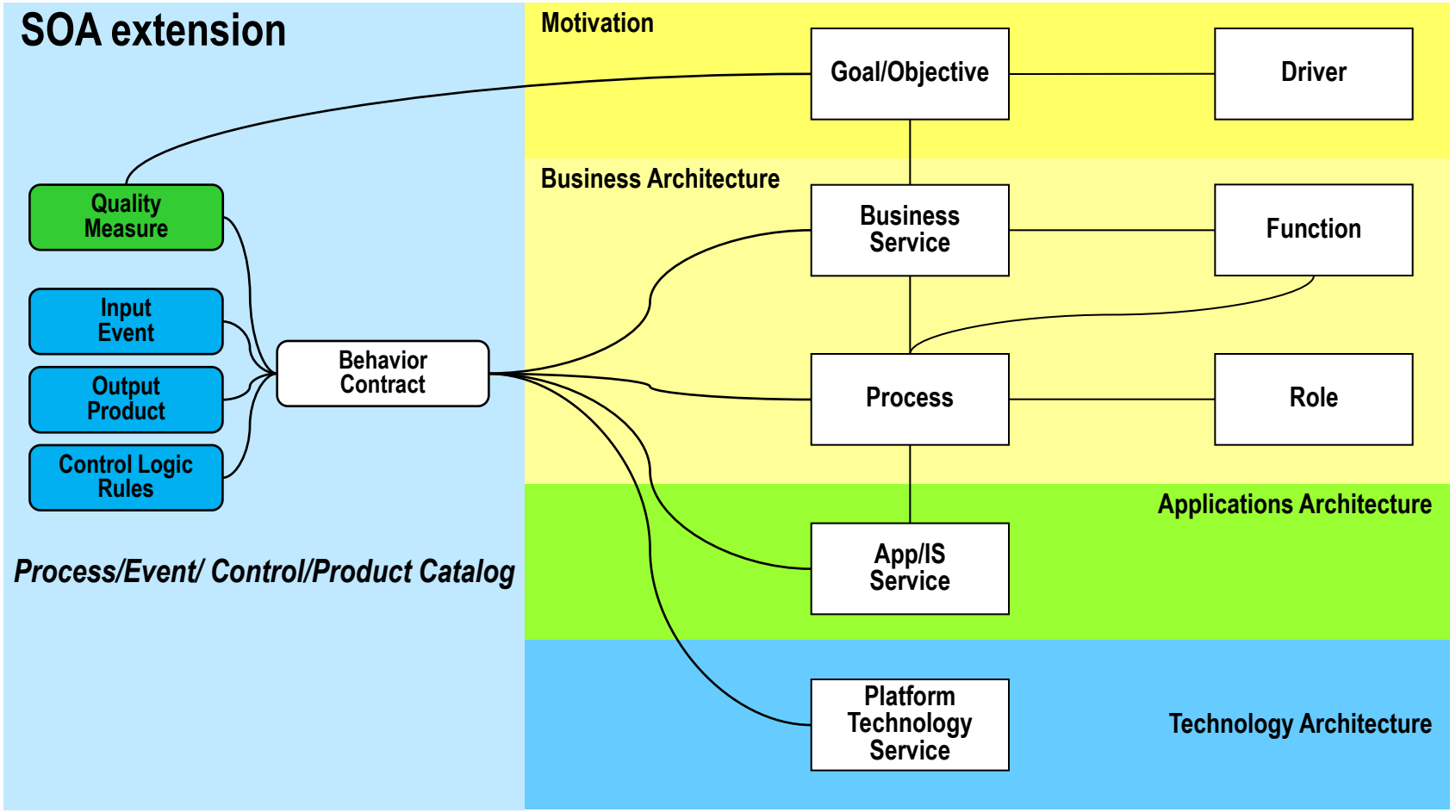


(1) A measure not associated with a quality is meaningless

(2) Control Logic/Rules = preconditions and post conditions

Adapting those extensions to form a simple SOA extension.

- Note how the SOA extension is already reflected in Process/Event/Control/Product Catalog.



Part 4: Harmonising TOGAF with Zachman Framework levels

Avancier

What	How	Where	Who	When	Why	Zachman levels
Passive structures	Behaviors	Places in space	Active structures	Points in time	Aims	
						Scope/Context
						Conceptual
						Logical
						Physical
						Configuration
						Instantiation

Enterprise Continuum mapped to Zachman Framework levels

Avancier

Generalization Idealization	Foundation (Universal)	Common Systems (Fairly generic)	Industry (Fairly specific)	Organization (Uniquely configured)	Zachman levels
Requirements and Context	Business and application service contracts				Scope/Context Concepts
Architecture Continuum	Logical or vendor-neutral specifications of ABBs.				Logical
Solution Continuum	Physical or implementation-specific nominations of SBBs				Physical
Deployed solutions					Instantiation

TOGAF mapped to Zachman Framework levels

Avancier

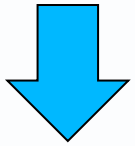
ADM Deliverable	Enterprise Continuum Enterprise Repository	Services & Building Blocks	Business domain entities	Applications domain entities	Data domain entities	Technology domain entities	System theory	Zachman levels
EA/Strategic Architecture			Function and organization hierarchies	Application portfolio catalog	Business data entity catalog	Technology services catalog (TRM)		Scope/Context
Architecture Req'ments Specification	Req'ments & Context Req'ments Repository	Business & Application Service Contracts	Business Services, Processes	App/IS Services			Required Behaviors	Conceptual
Architecture Definition Document	Architecture Continuum Architecture Repository	Architecture Building Blocks	Functions, Roles	Logical App Components	Logical data entities, Data components	Logical Technology Components	Logical structure description	Logical
Architecture Road Map	Solutions Continuum Solutions Repository	Solution Building Blocks	Organization Units, Actors	Physical App Components	Physical Data Components	Physical Technology Components	Physical structure description	Physical
Architecture Change Requests	Deployed Solutions		Identity Management	IT Configuration Management (CMDB)			Concrete system realization	Configuration
			Business & IT Operations					Instantiation

Part 5: Harmonising ArchiMate with TOGAF

Avancier

TOGAF's general principles (repeat)

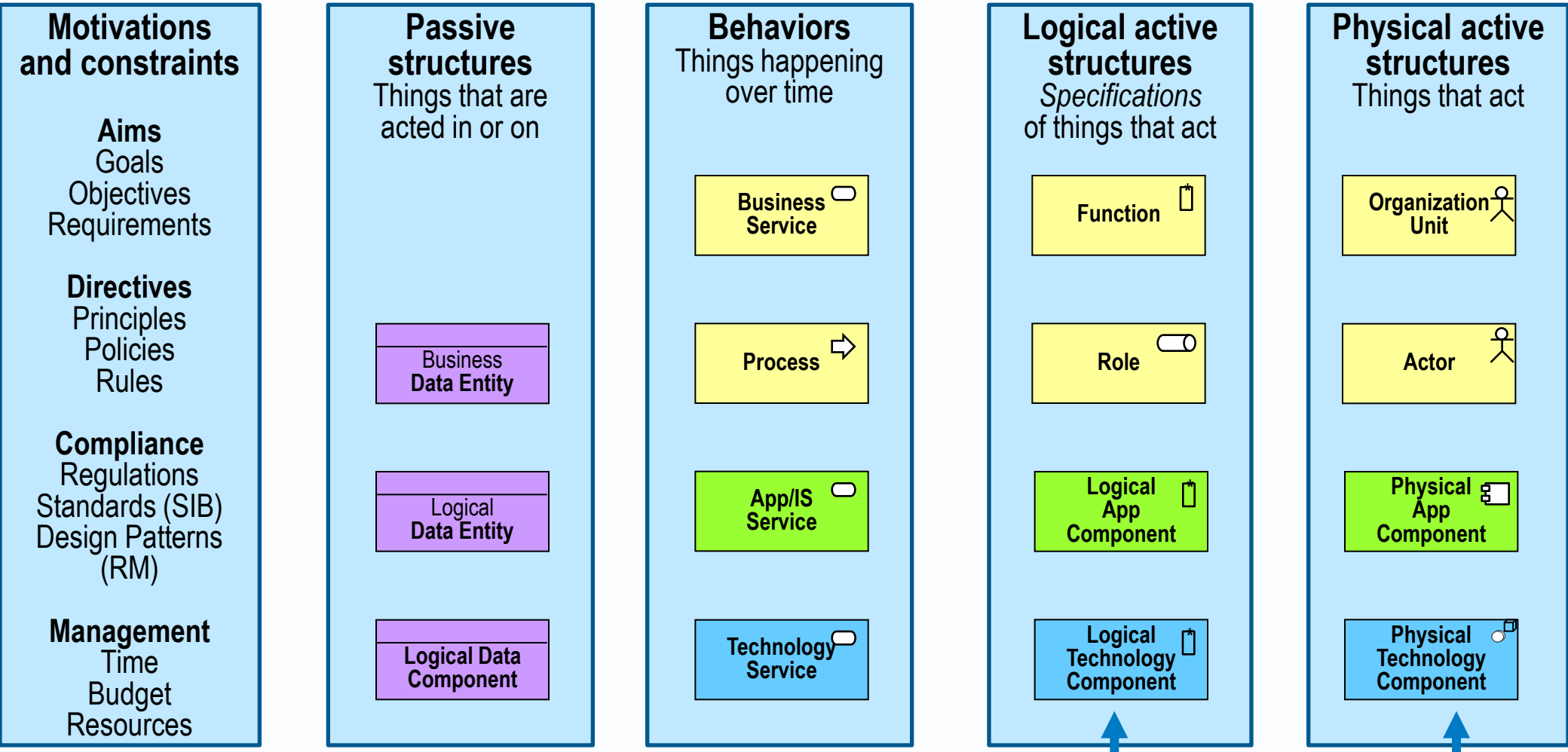
- Enterprise architects should separate logical specifications from physical realizations.



Motivations and constraints	Passive structures	Behaviors	Logical structures	Physical structures
	Things that are acted on	Things that happen	Specifications of thing that act	Things that act
Drivers Goals/Objectives	Business Data Entities	Business Services, Processes	Functions, Roles	Organization Units, Actors
	Logical Data Entities	IS Services	Logical Application Components	Physical Application Components
		Platform Services	Logical Technology Components	Physical Technology Components

TOGAF is about service-oriented implementation-independent specification of business systems

Avancier



And separates **logical ABBs** from **physical SBBs**

Physical things in EA are still "considerably abstracted from implementation"

The ArchiMate standard is not a universal truth to which TOGAF must conform

Avancier

Peculiar structure/behavior distinction

- In modelling activity systems, all active structures are definable in terms of the activities they perform.
- E.g. active objects in UML are structures, defined by the operations they perform.
- Curiously, ArchiMate regards any described group of activities as a behaviour.

Peculiar position of event

- In modelling human and computer activity systems, all behaviors are event-driven
- Curiously, ArchiMate associates events with processes, but not services.

No differentiation of logical structures from physical/real ones

- In modelling activity systems, a role lists activities to be performed by actors playing that role.
- Actors are hired, bought or built on the basis they can perform those activities.
- E.g. a class in UML lists operations to be performed by objects of hat class.
- Roles and functions do not do things; they only specify what actors and organization units do.

Confusion over what a function is

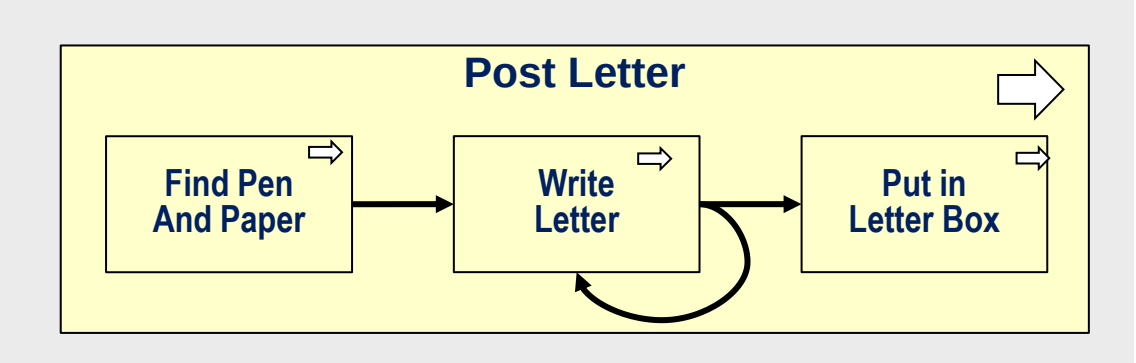
- ArchiMate appears to define business functions exactly as TOGAF does.
- But in the application and technology domains, functions sometimes appear to be processes, and other times appear to be components.

Component-bound interfaces

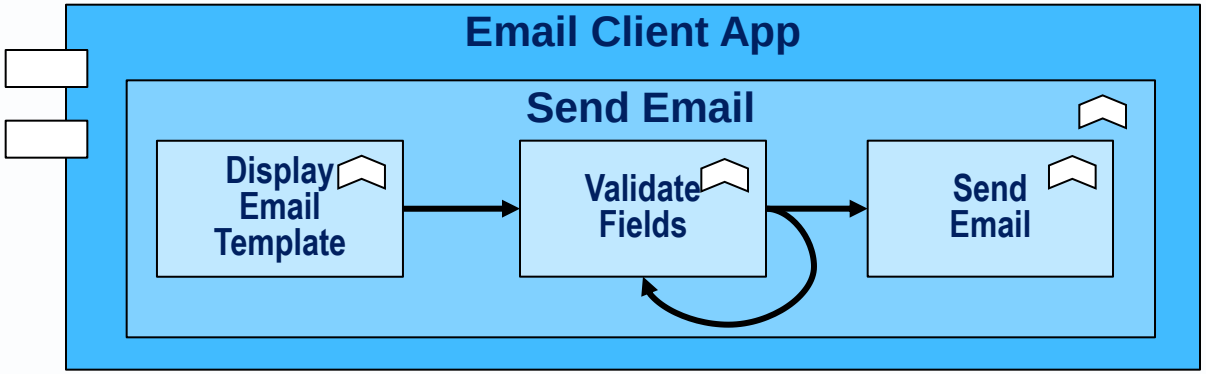
- ArchiMate doesn’t allow interfaces to be decoupled from components.
- Or allow an interface to identify services offered more than one component.

The function / process confusion

- ArchiMate interprets "behavior" differently from UML
  - People misunderstand "business function" and confuse function with process.
- Where some use process symbols in the business layer



- Some use functions symbols application layer



ArchiMate generic meta model - simplified

Avancier

	Behavior	Structure
External view	Services	Interfaces
Internal view	Processes	Active Structures

ArchiMate adapted to fit TOGAF

Avancier

	Behavior	Structure
External view	Services	Interfaces
Internal view	Processes	Logical Structures
		Physical Active Structures

- Logical structures aggregate properties (e.g. abilities and behaviors) that physical structures are expected to realise.
- Physical structures are nominated to realise those properties (e.g. perform those behaviors.)

TOGAF version of the same

Avancier

	Behavior	Structure	Generic meta model
External view	Services	Interface / Boundary / Service portfolio	**Required behaviors**
Internal view	Processes	Logical Architecture Building Blocks	**Logical structures**
		Physical Solution Building Blocks	**Physical structures**

- Logical structures aggregate properties (e.g. abilities and behaviors) that physical structures are expected to realise.
- Physical structures are nominated to realise those properties (e.g. perform those behaviors.)

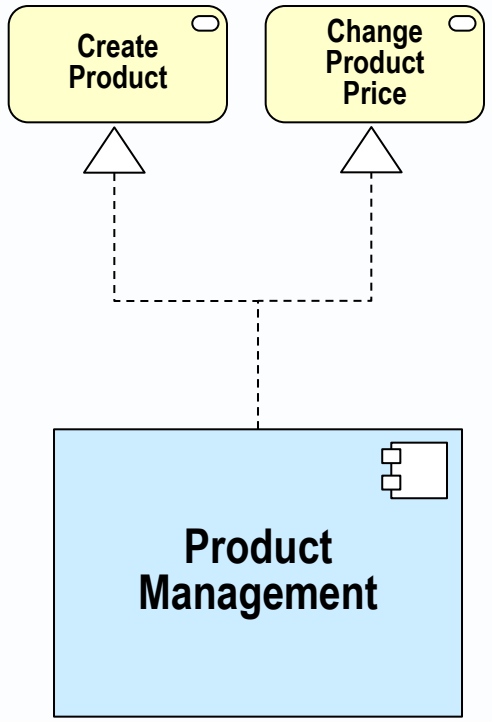
Part 6: Harmonising TOGAF with ArchiMate

- ArchiMate positions intermediate entities on the relationships between services and components.

	ArchiMate's core entities and relationships		Omitting intermediate entities
External behavior	**Services**		**Services**
	<are realised by>	<are assigned from>	
Intermediate entities	**Functions**	**Interfaces**	<are realised by>
	<are assigned from>	<are part of>	
Internal active structure	**Components**		**Components**

TOGAF promotes abstraction

- TOGAF defines ABBs primarily in terms of external services (or service clusters as exemplified in the TRM)
- It shows little or no interest in the "internal behaviors" of application or technology components, or their interfaces.

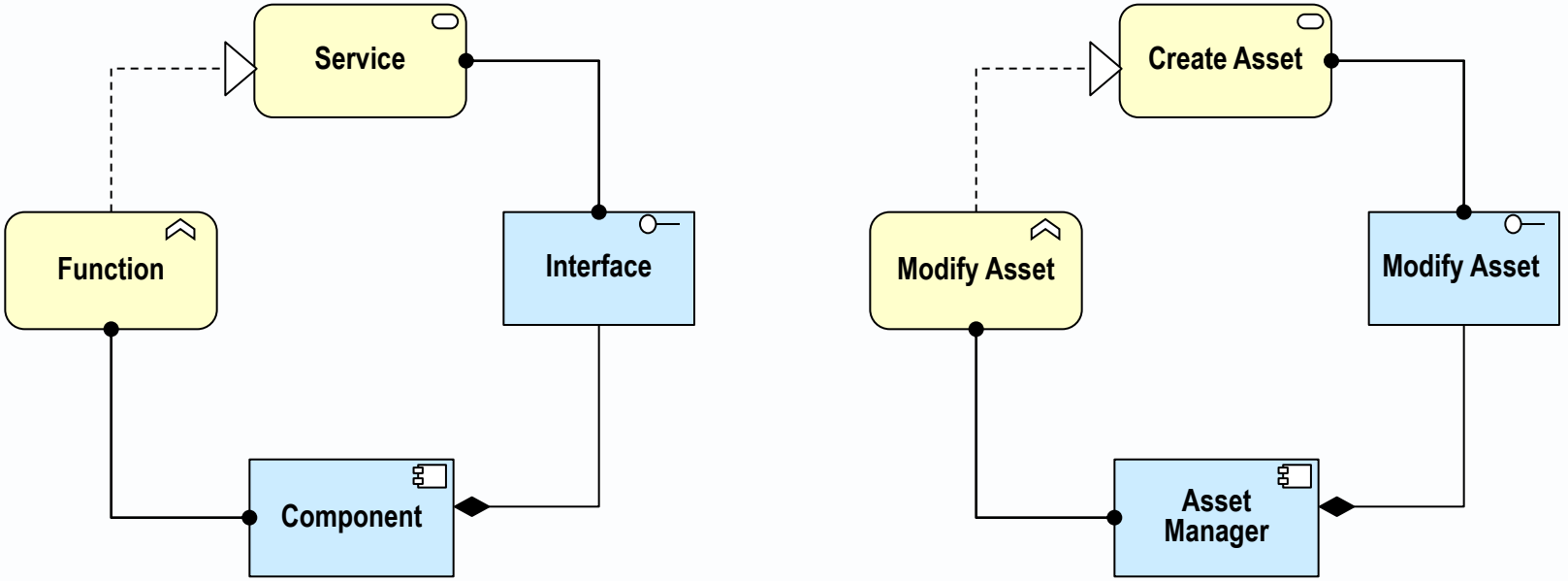


Technology Service Catalog (aka TRM)

User Interface Services	Transaction Processing Services	Operating System Services	Software Engineering Services
Graphical Client/Server services	Starting a transaction	Kernel Operations	Programming Language services
Display Objects services	Co-ordination of recoverable resources in a transaction	Command Interpreter and Utility services	Object Code Linking services
Window Management services	Committing or rolling back transactions	Batch Processing services	CASE Environment and Tools services
Dialogue Support services	Controlling timeouts on transactions	File and Directory Synchronization	Graphical User Interface (GUI) Building services
Printing services	Chaining transactions together		Scripting Language services
Computer-Based Training and Online Help services	Monitoring transaction status		Language Binding services
Character-Based services			Run-Time Environment services
			App Binary Interface services
Graphics and Imaging Services	**Data Management Services**	**Network Services**	**OO Provision of Services**
Graphics services	Data Dictionary/Repository services	Electronic Mail services	Object Request Broker (ORB) services
Graphical Object Management services	Database Management System (DBMS) services	Distributed Data services	Implementation Repository services
Drawing services	OO Database Management System (OODBMS) services	Distributed File services	Installation and Activation services
Imaging functions	File Management services	Distributed Name services	Interface Repository services
	Query Processing functions	Distributed Time services	Replication services
International Operation Services	Screen Generation functions	Remote Process (Access) services	Common Object services
Character Sets and Data Representation services	Report Generation functions	Remote Print Spooling and Output Distribution services	Change Management services
Cultural Convention services	Networking/Concurrent Access functions	Enhanced Telephony functions	Collections services
Local Language Support services	Warehousing functions	Shared Screen functions	Concurrency Control services
		Video-Conferencing functions	Data Interchange services
		Broadcast functions	Event Management services
		Mailing List functions	Externalization services
Data interchange services	**Location and Directory Services**	**System and Network Management Services**	Licensing services
Document Generic Data Typing and Conversion services	Directory services	User Management services	Lifecycle services
Graphics Data Interchange services	Special-Purpose Naming services	Configuration Management (CM) services	Naming services
Specialized Data Interchange services	Service Location services	Performance Management services	Persistent Object services
Electronic Data Interchange services	Registration services	Availability and Fault Management services	Properties services
Fax services	Filtering services	Accounting Management services	Query services
Raw Graphics Interface functions	Accounting services	Security Management services	Relationship services
Text Processing functions		Print Management services	Security services
Document Processing functions	**Security Services**	Network Management services	Start-Up services
Publishing functions	System Entry Control services	Backup and Restore services	Time services
Video Processing functions	Security Management services	Online Disk Management services	Trading services
Audio Processing functions	Audit services	License Management services	
Media Synchronization functions	Access Control services	Capacity Management services	
Multimedia Processing functions	Non-Repudiation services	Software Installation services	
Information Presentation and Distribution functions	Trusted Recovery services	Trouble Ticketing services	
Hypertext functions	Encryption services		
	Trusted Communication services		

ArchiMate's two relationships between service and component

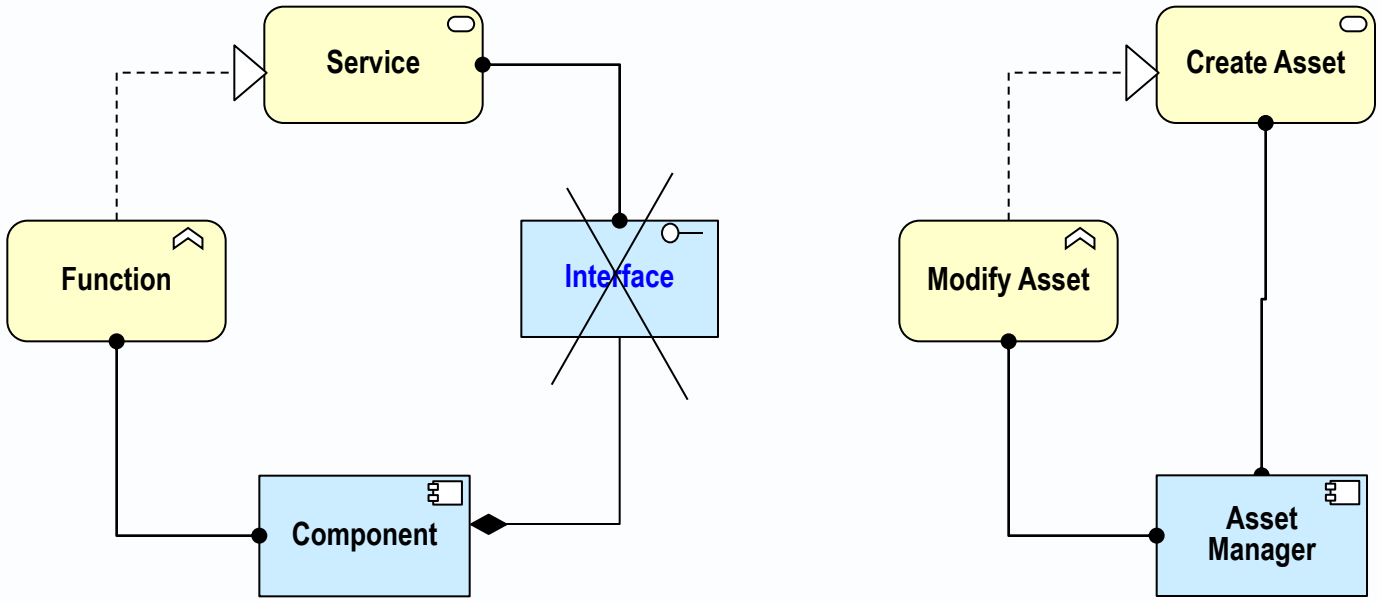
Service <realised by> Function <assigned from> Component
Service <assigned from> Interface <part of> Component



However, the intermediate entities can be omitted and ArchiMate's relationship precedence can be applied.

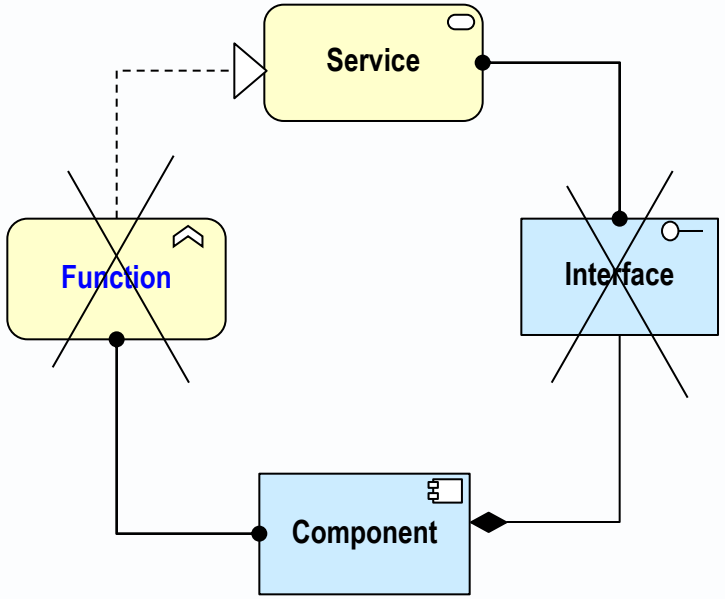
Omitting Interfaces

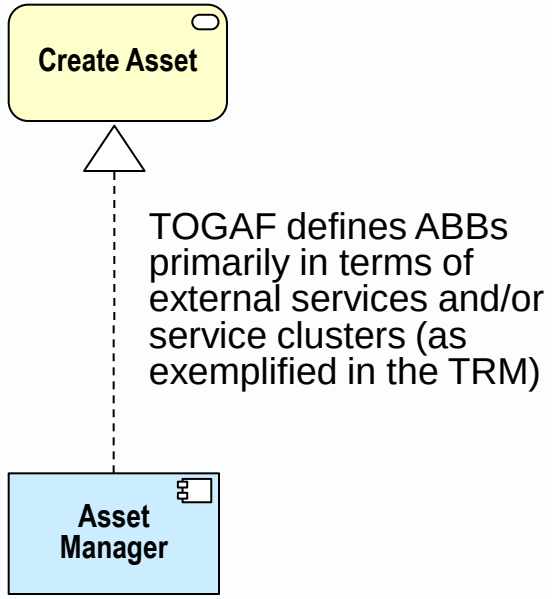
- TOGAF does not define a component's interfaces
  - (other than as a service portfolio, perhaps very sketchy)



Omitting Functions

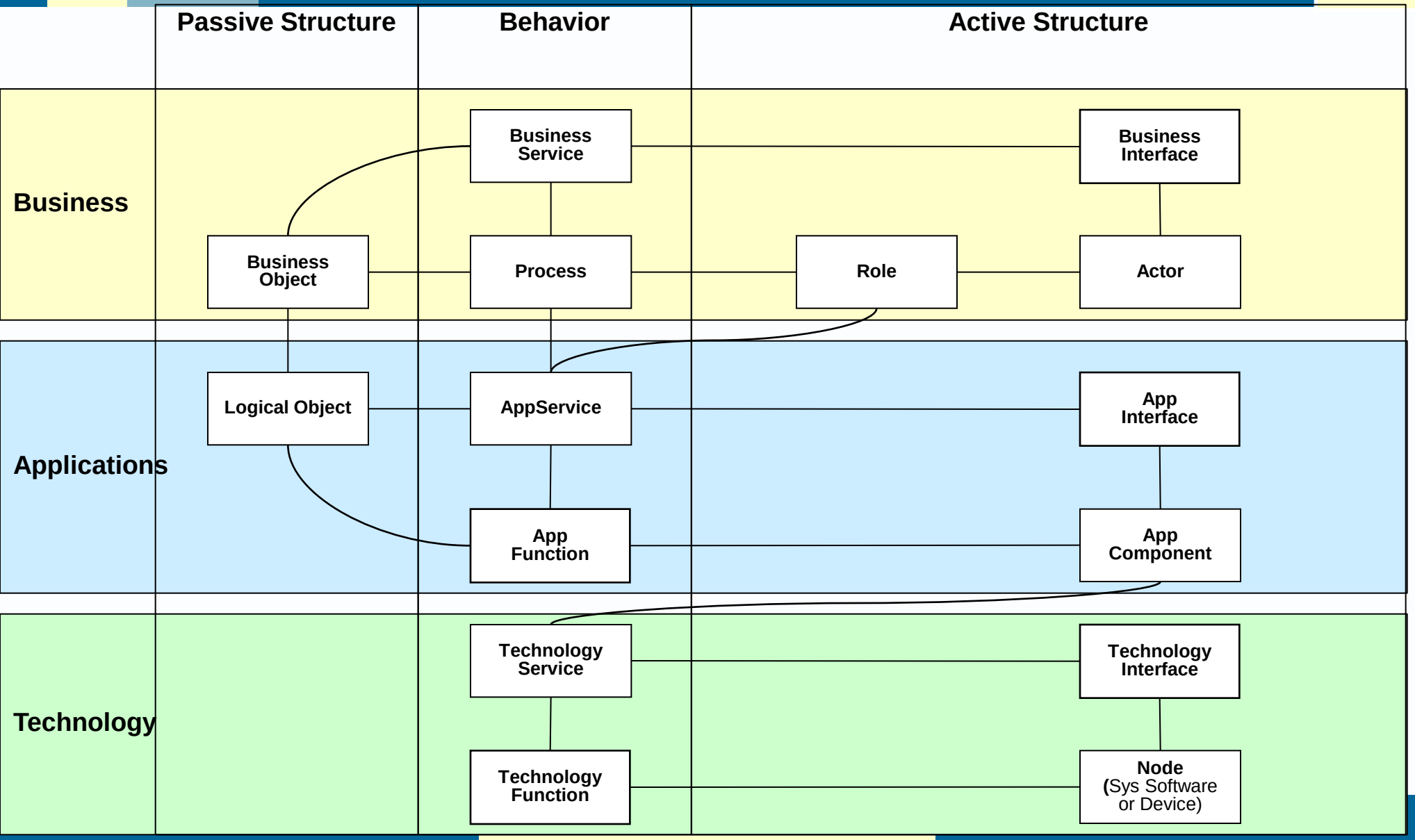
- TOGAF has little interest in the "internal behaviors" of application and technology components
  - (except in so far as they might be specified in service contracts)



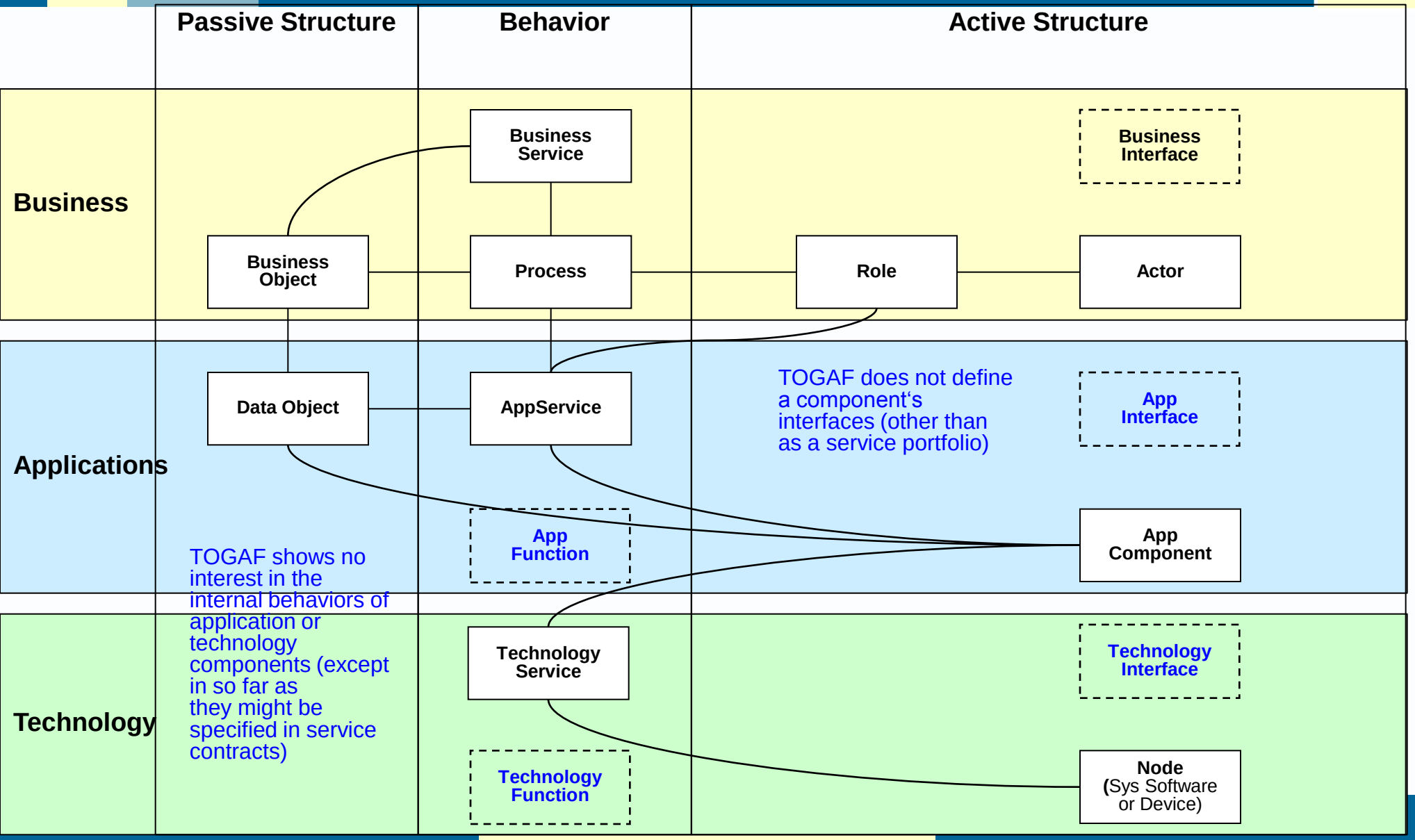


Starting from this part of the ArchiMate meta model

Avancier

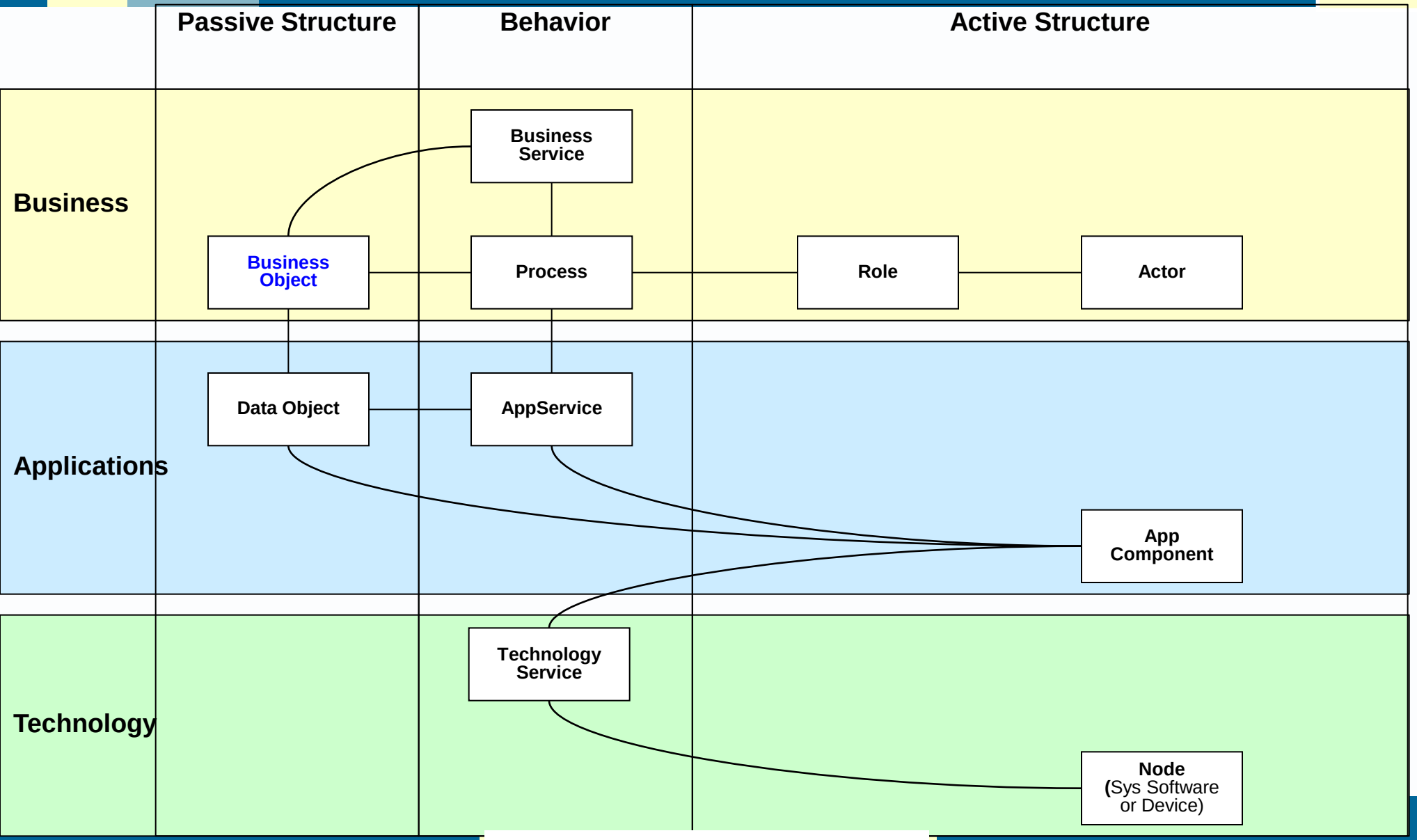


Removing five entities not in TOGAF

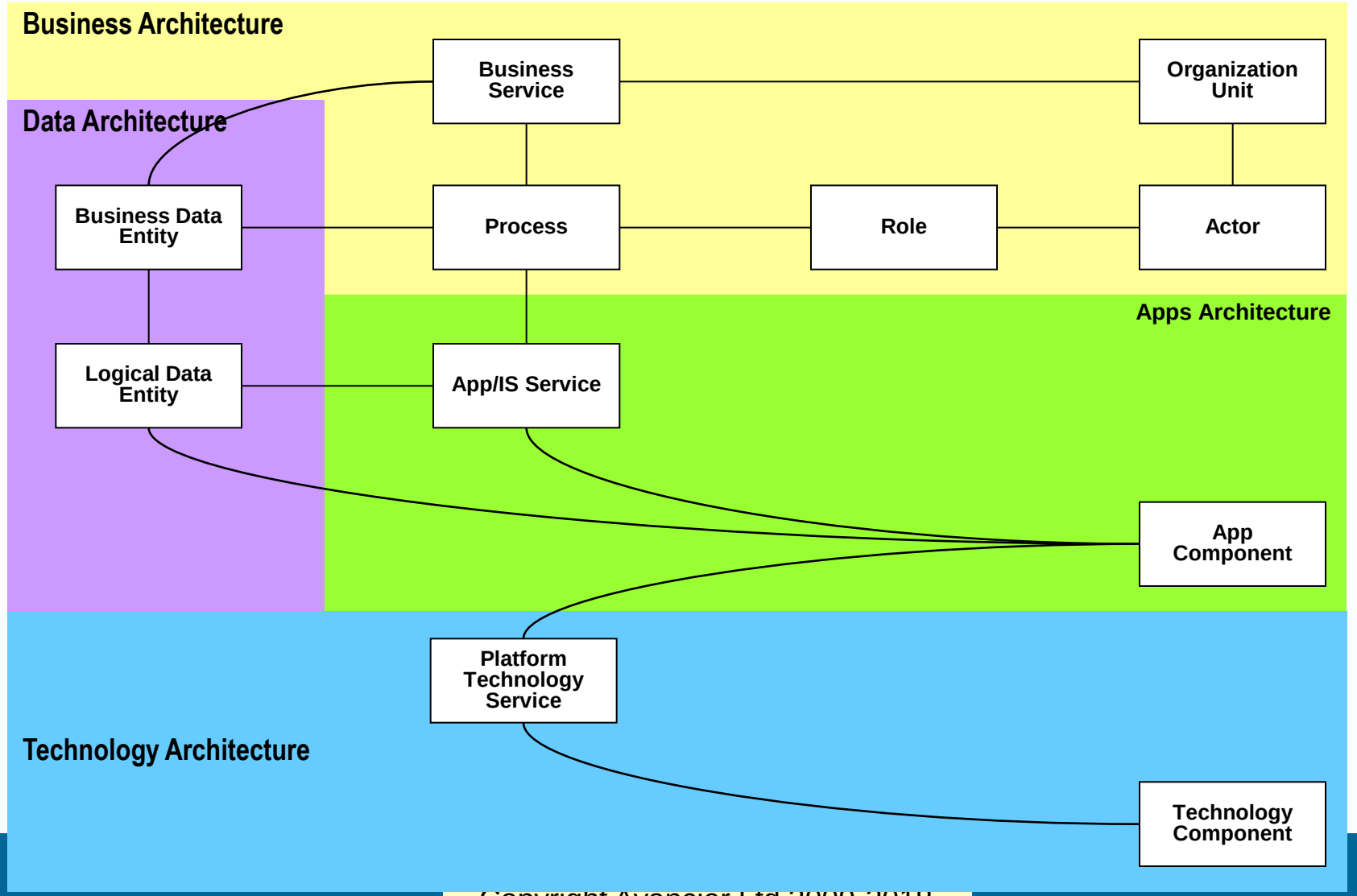


Retaining one entity not in TOGAF

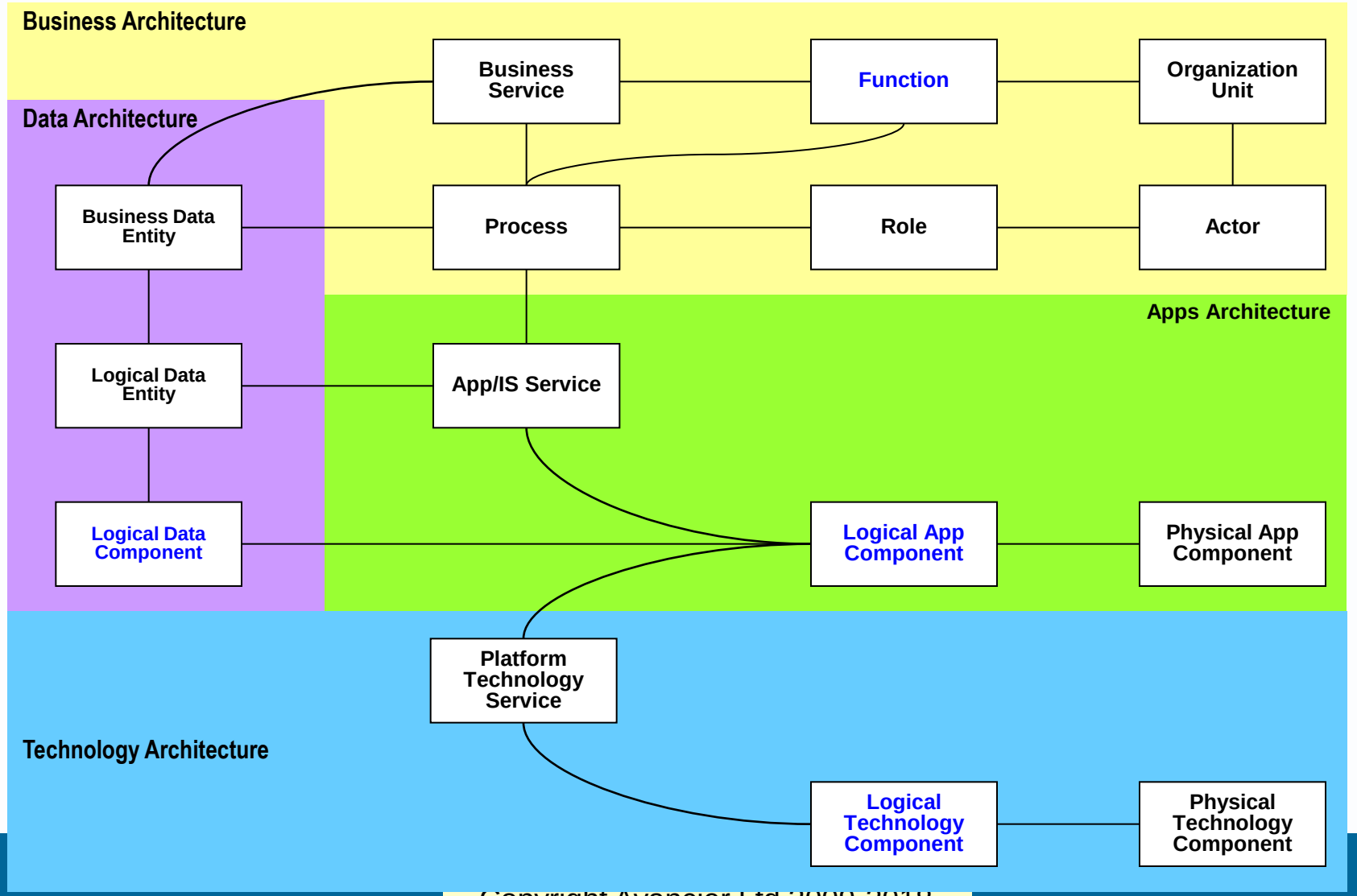
Avancier

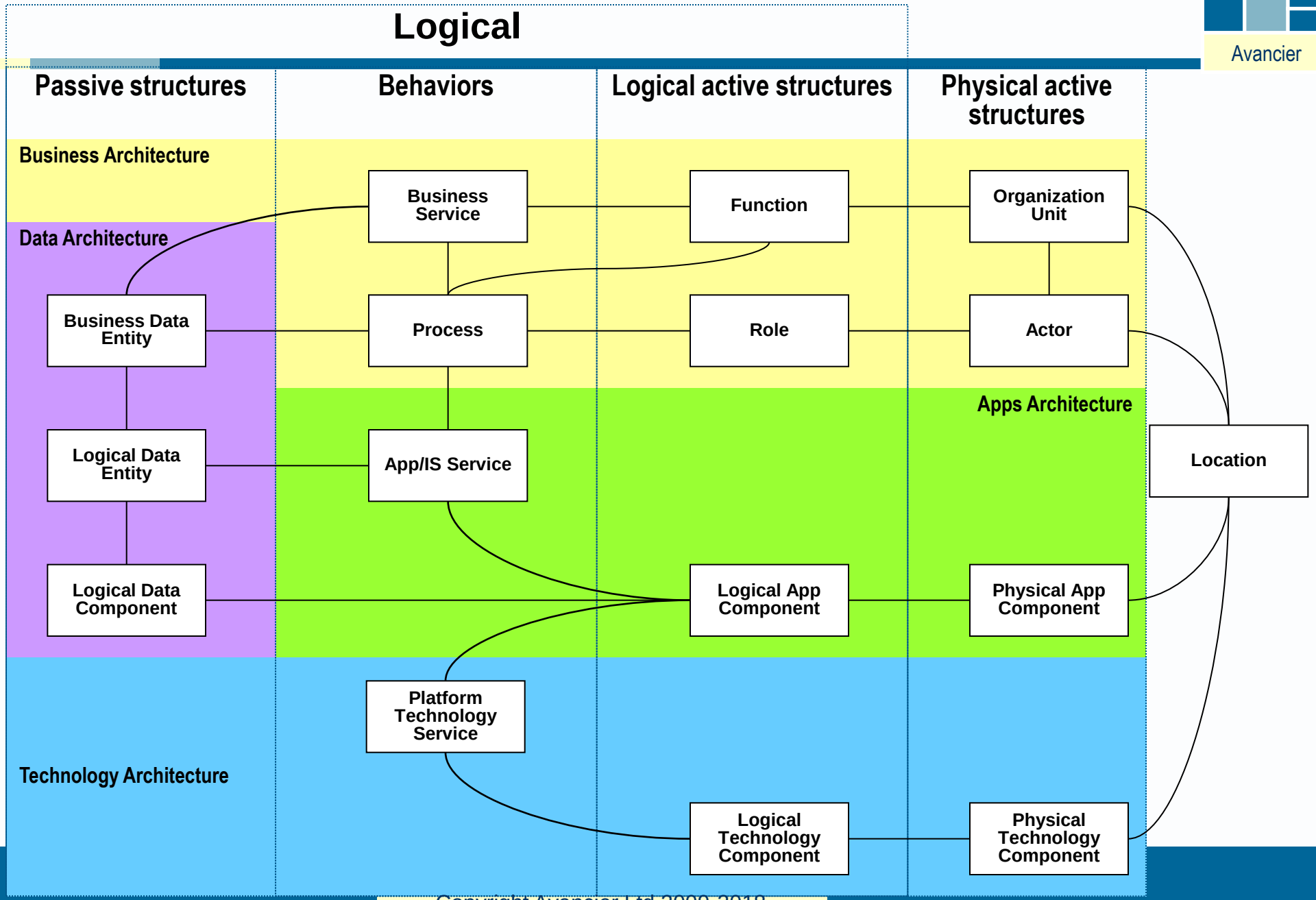


Rename as the entities as in TOGAF (tempted to stop here?)



Adding four logical structure components found in TOGAF

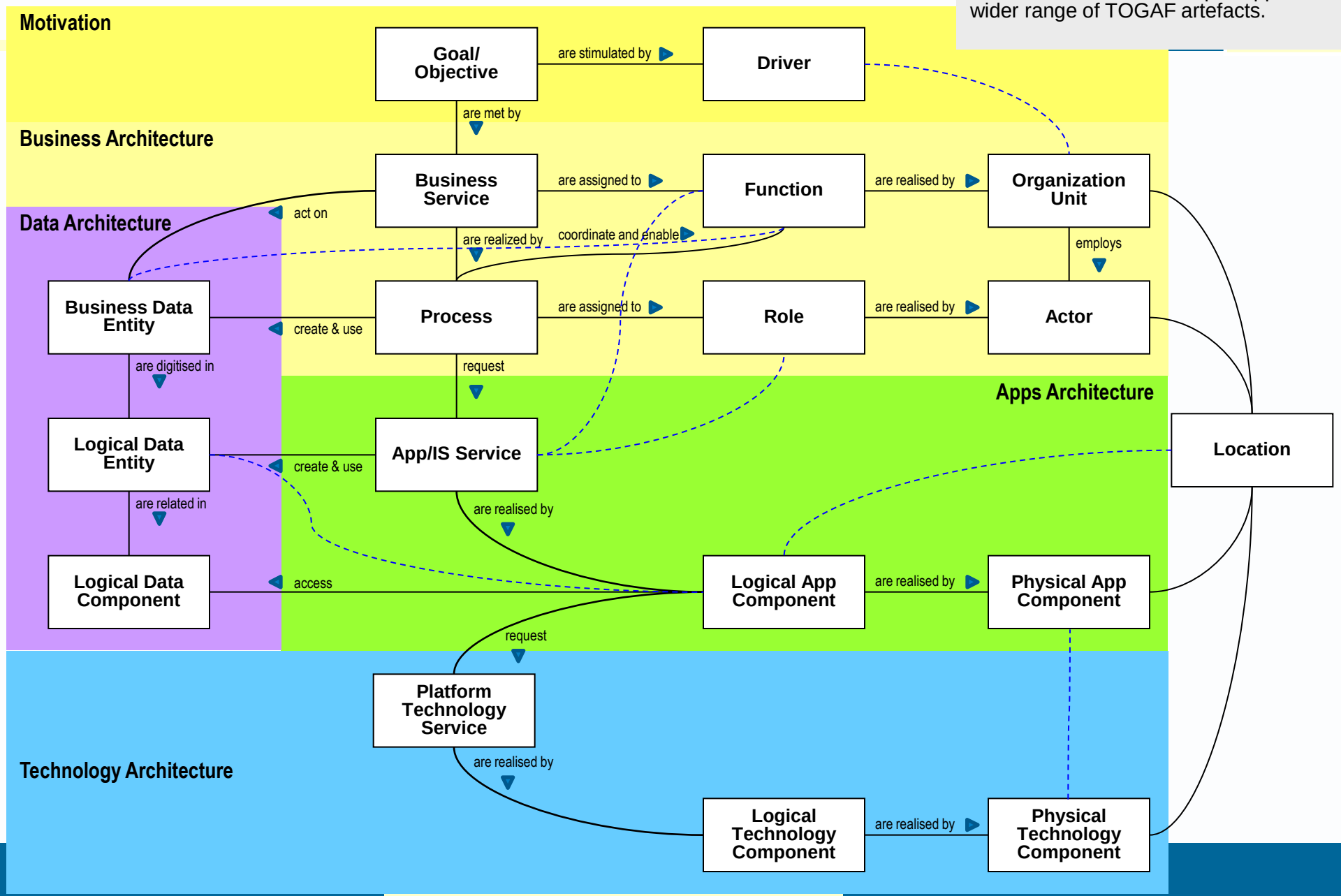




A TOGAF meta model abstracted from ArchiMate

The solid line relationships are found in the ArchiMate standard (if I have it right).

The dotted line relationships support the wider range of TOGAF artefacts.



Part 7: Radically changing TOGAF to align with ArchiMate?

- The Open Group promotes service-oriented, implementation-independent specification of business systems.

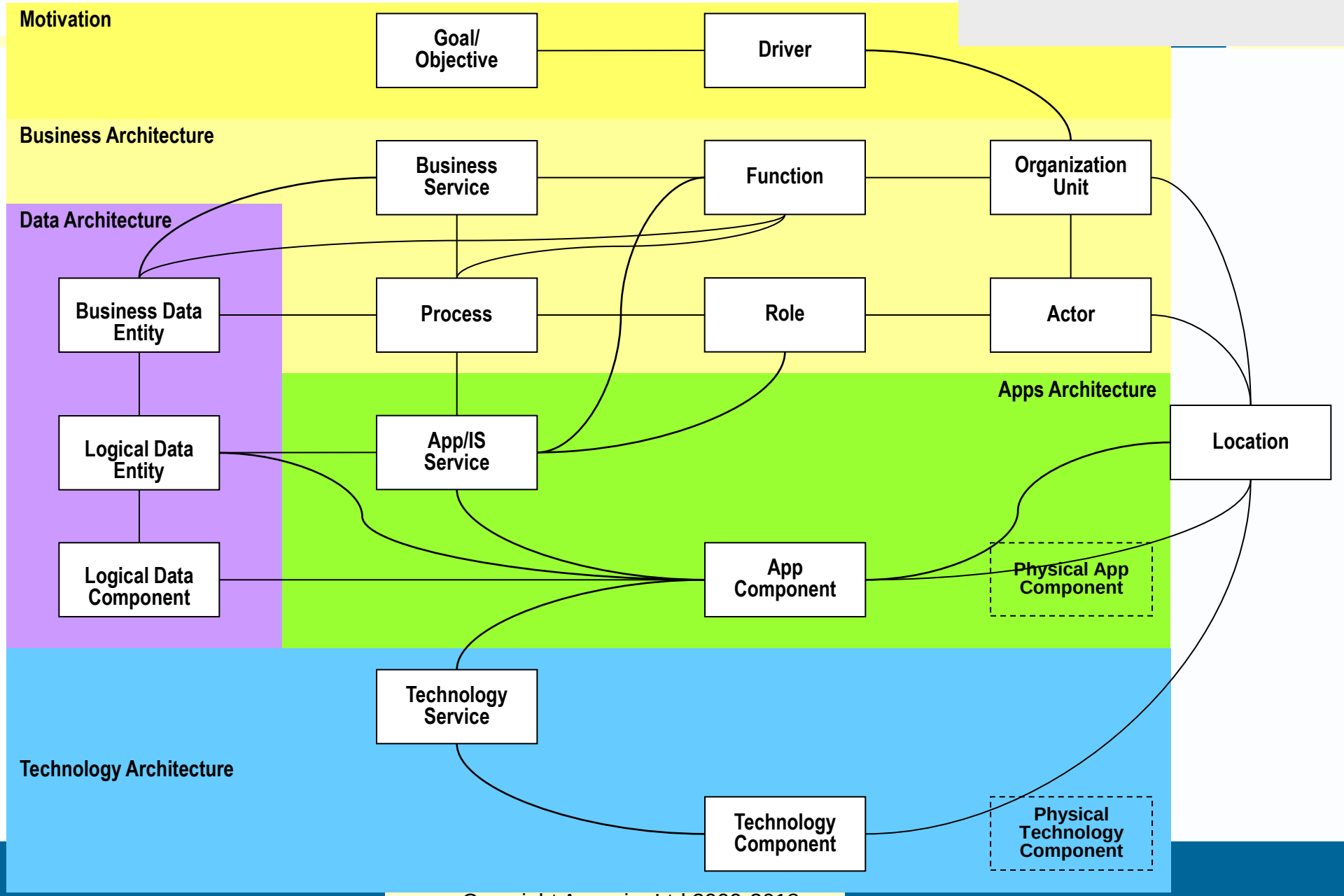
- So, TOGAF separates logical architecture buildings blocks from physical solution building blocks.
- This separation runs all the way through TOGAF's
  - architecture development method.
  - meta model,
  - deliverables,
  - enterprise continuum and
  - enterprise repository.

However…

- Modelling languages are agnostic about the level of abstraction to be used in system specification.
- So what if we remove the separation of logical architecture components from physical solution components?
- Even though it runs all through TOGAF!

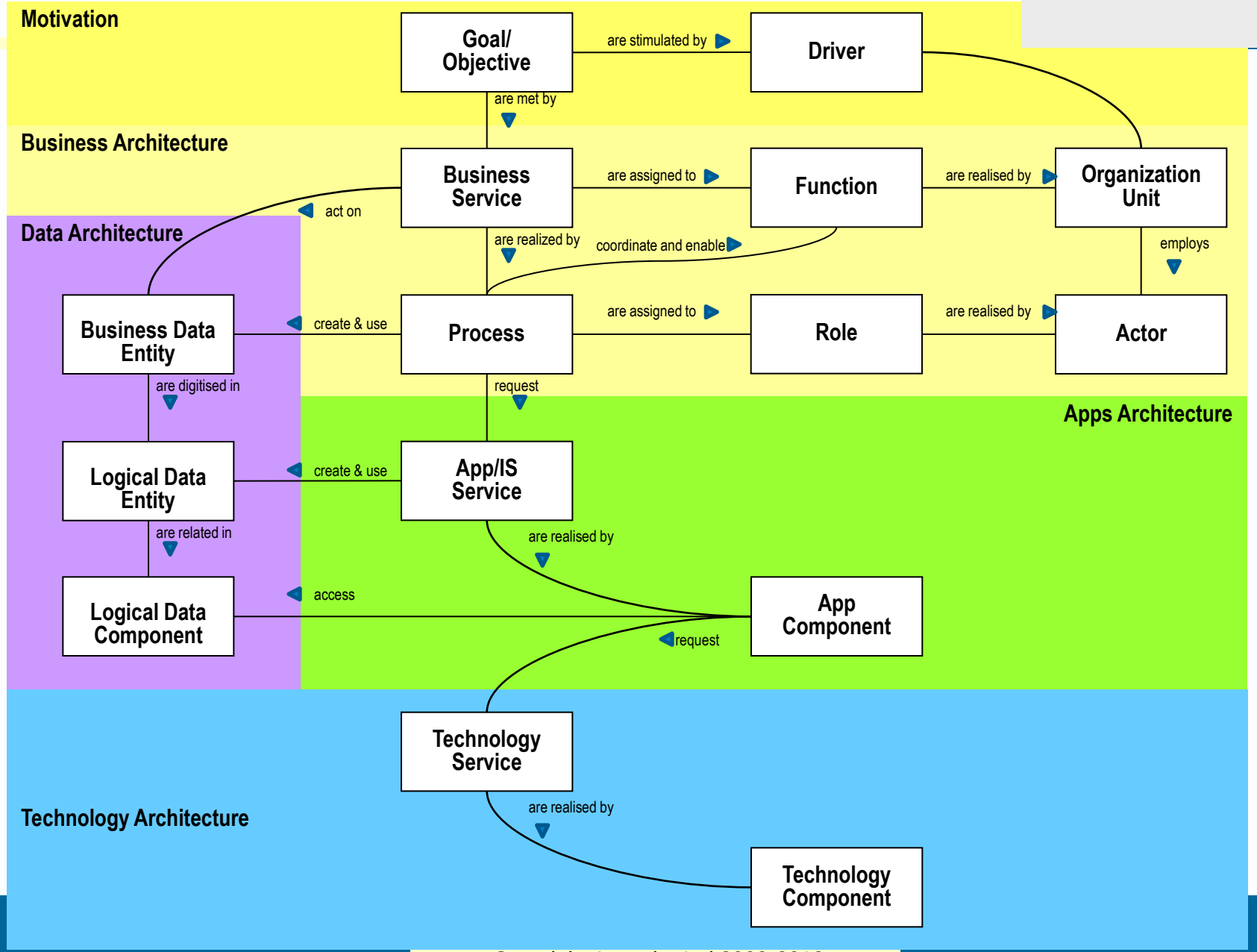
A meta model without the log/phys component pairs

Note again: most entities may be recursively composed and decomposed, and all relationships are many to many.



Further simplified and annotated

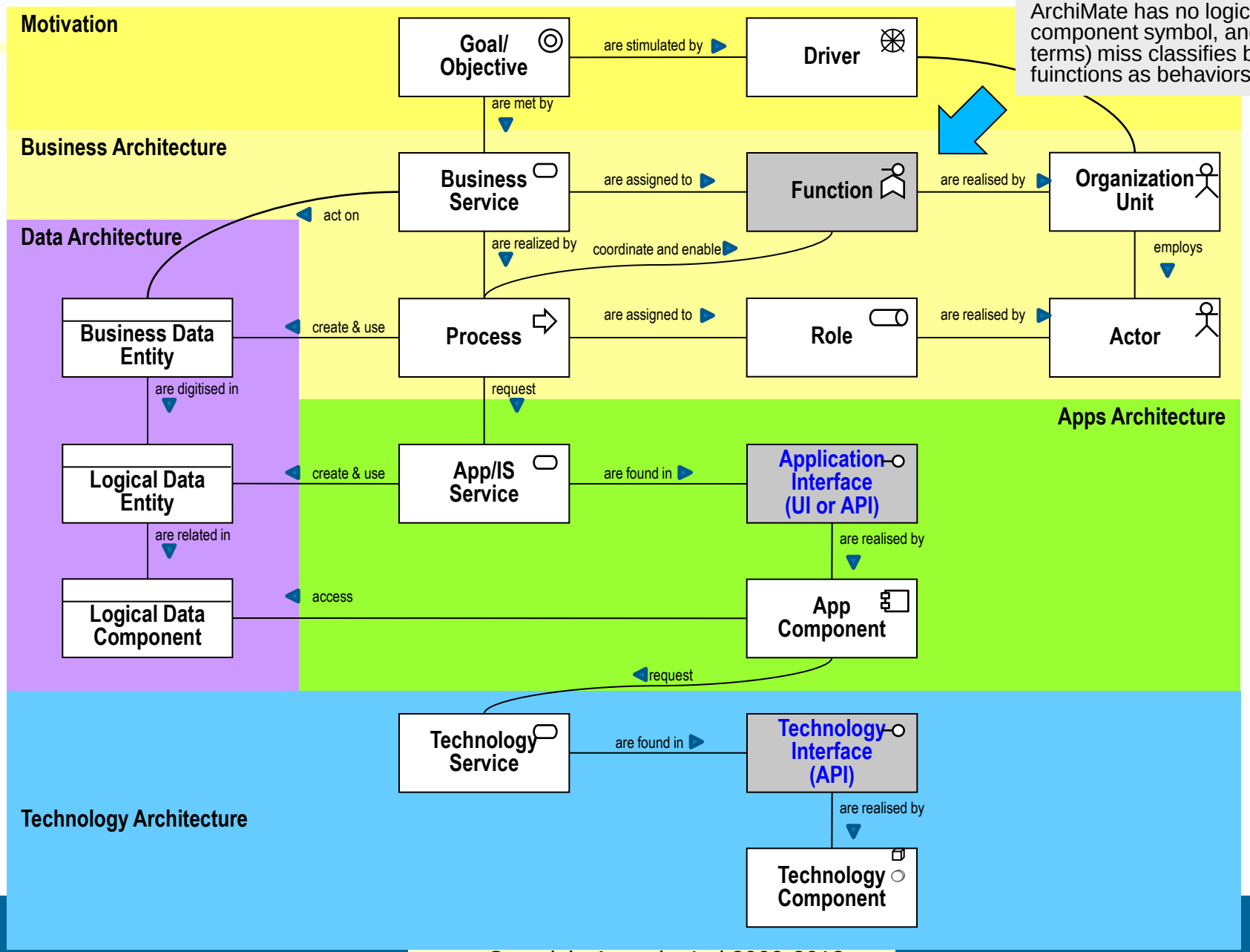
Note again: most entities may be recursively composed and decomposed, and all relationships are many to many.



A more ArchiMate-compatible TOGAF meta model

Note again: most entities may be recursively composed and decomposed, and all relationships are many to many.

ArchiMate has no logical structure/ component symbol, and so (in TOGAF terms) miss classifies business fuinctions as behaviors.



Radically realigned framework

Avancier

ADM Deliverable	Enterprise Continuum Enterprise Repository	Services & Building Blocks	Business domain entities	Applications domain entities	Data domain entities	Technology domain entities
EA/Strategic Architecture			Function and organization hiearchies	Application portfolio catalog	Business data entity catalog	Technology services catalog (TRM)
Architecture Req'ments Specification	Req'ments & Context Req'ments Repository	Business & Application Service Contracts	Business Services, Processes	App/IS Services		
Architecture Definition Document	Architecture Continuum Architecture Repository	Architecture Building Blocks	Functions, Roles	Application Interfaces (UI or API)	Logical data entities, Data components	Technology Interfaces (API)
Architecture Road Map	Solutions Continuum Solutions Repository	Solution Building Blocks	Organization Units, Actors	App Components	Data Components	Technology Components
Architecture Change Requests	Deployed Solutions		Identity Management	IT Configuration Management (CMDB)		
			Business & IT Operations			

But the impact on TOGAF is fundamental

- Since the separation of
  - logical “architecture buildings blocks” from
  - physical “solution building blocks.”

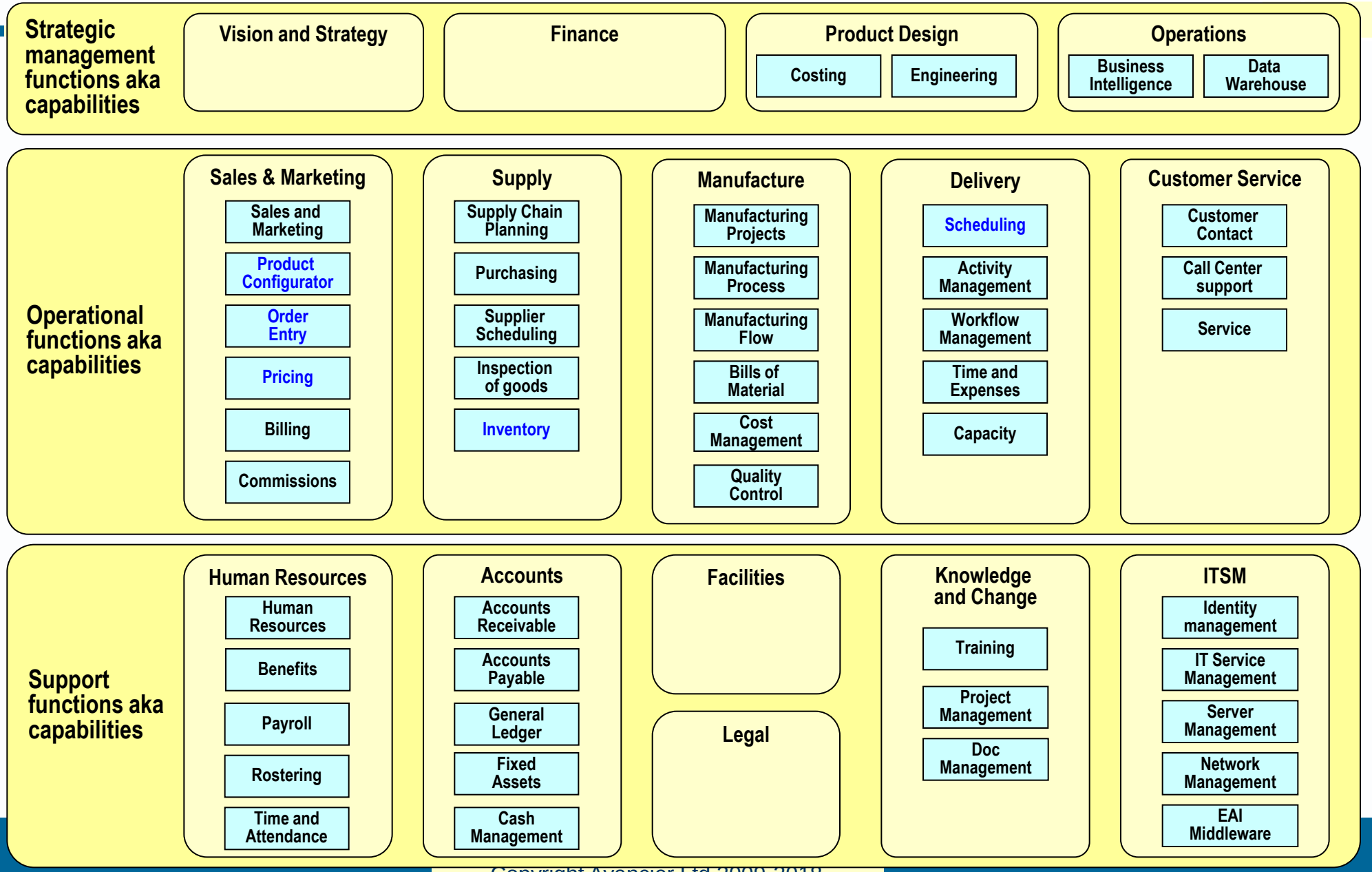
- runs all the way through TOGAF’s
  - ADM,
  - deliverables,
  - meta model,
  - enterprise continuum
  - enterprise repository.

Part 8: Harmonising TOGAF with Capability-Based Planning

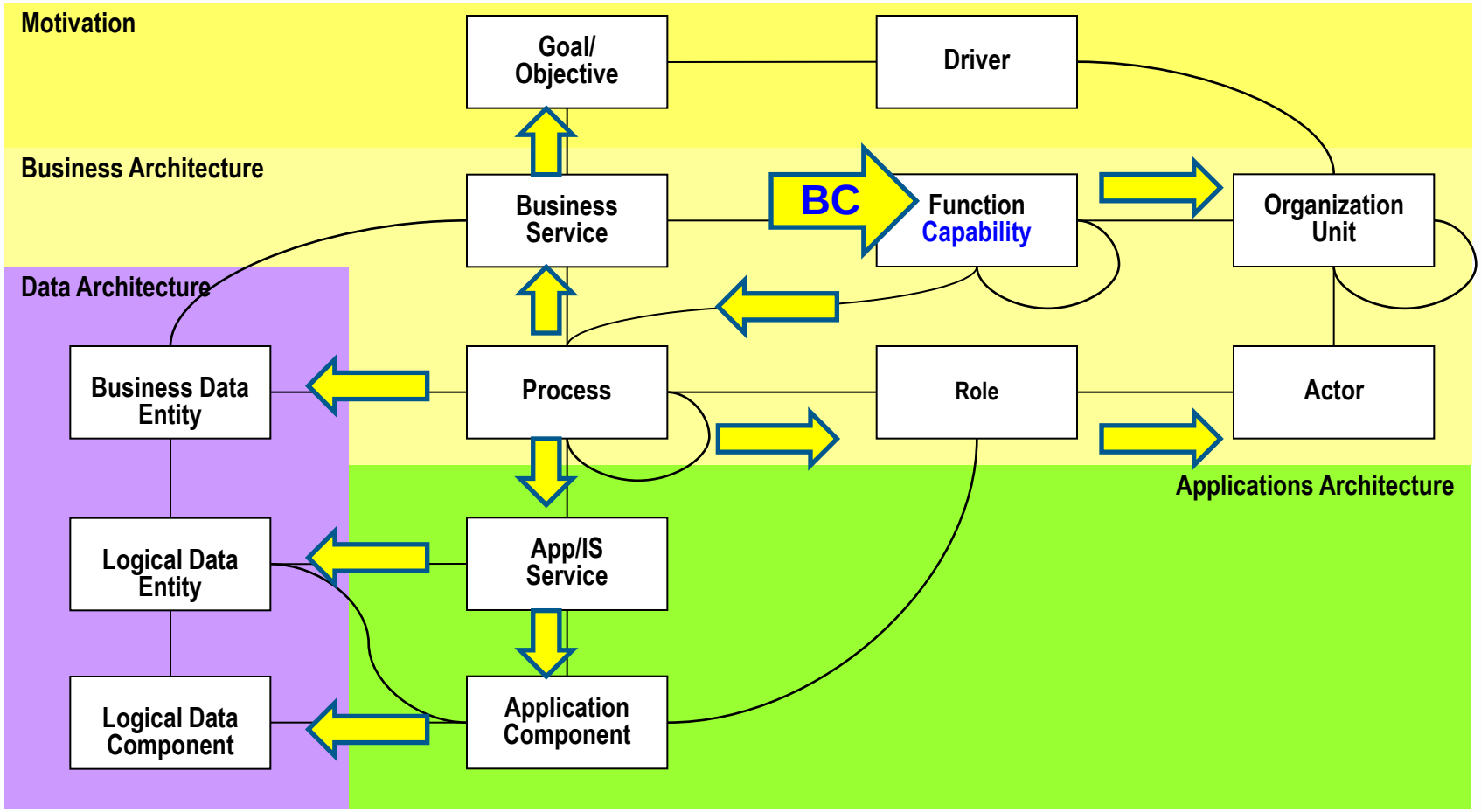
Avancier

Functional Decomposition Diagram (as in TOGAF 9.1)	Capability Map
Shows on a single page the organization capabilities relevant to the architecture to be defined and governed. Helps to quickly model the organization's capabilities without being dragged into debate on how the organization does it. Given a basic diagram, it is possible to layer heat-maps on top of it to show scope and decisions. For example, the capabilities to be implemented in different phases of a change program.	Shows on a single page the organization capabilities relevant to the architecture to be defined and governed. Helps to quickly model the organization's capabilities without being dragged into debate on how the organization does it. Given a basic diagram, it is possible to layer heat-maps on top of it to show scope and decisions. For example, the capabilities to be implemented in different phases of a change program.

Applications mapped to a two-level function/capability hierarchy



Business Capability in the meta model



Part 9: Harmonising TOGAF with Value Streams

Avancier

Process Flow Diagram (as in TOGAF 9.1)	Value Stream Diagram
Given a product or service of value to be delivered, this presents the necessary activities/steps in sequence. It may show for the process and each step • Trigger events • Outputs from processes, and • Controls/rules (pre and post conditions). It may use swim lanes to represent owners, roles or resources associated with process steps. It can help subject specialists to describe ‘‘how the job is done’’ for a particular function.	Given a product or service of value to be delivered, this presents the necessary activities/stages in sequence. It may show for the stream and each stage • Objectives and Roles involved • Entry criteria: including trigger events • Exit criteria: including products that are generated. It may associate owners, roles and capabilities with each value stage. It can help subject specialists to describe ‘‘how the job is done’’ for a particular capability.

Business scenarios (cf. value streams)

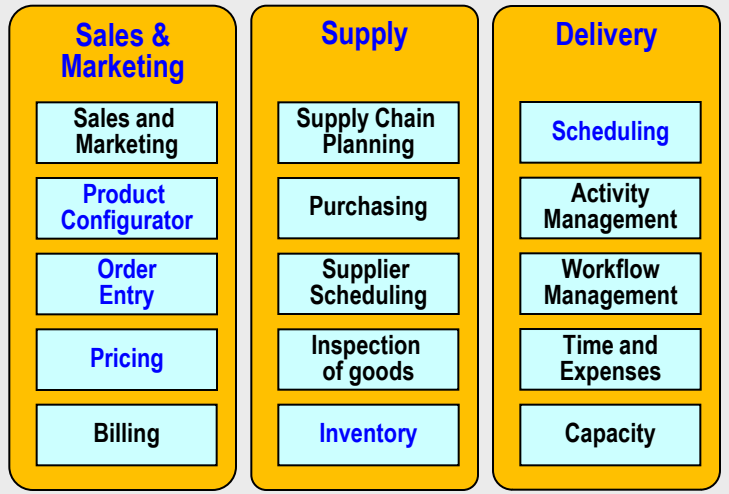
- EA looks to standardise, integrate, optimise and innovate in business processes and roles that create and use business data.

- TOGAF recommends starting from the products and services needed to achieve business goals.
- And defining the “architecture vision” by documenting business scenarios that deliver those products and services.
- A business scenario outlines a business process and the roles of human and computer actors in it.

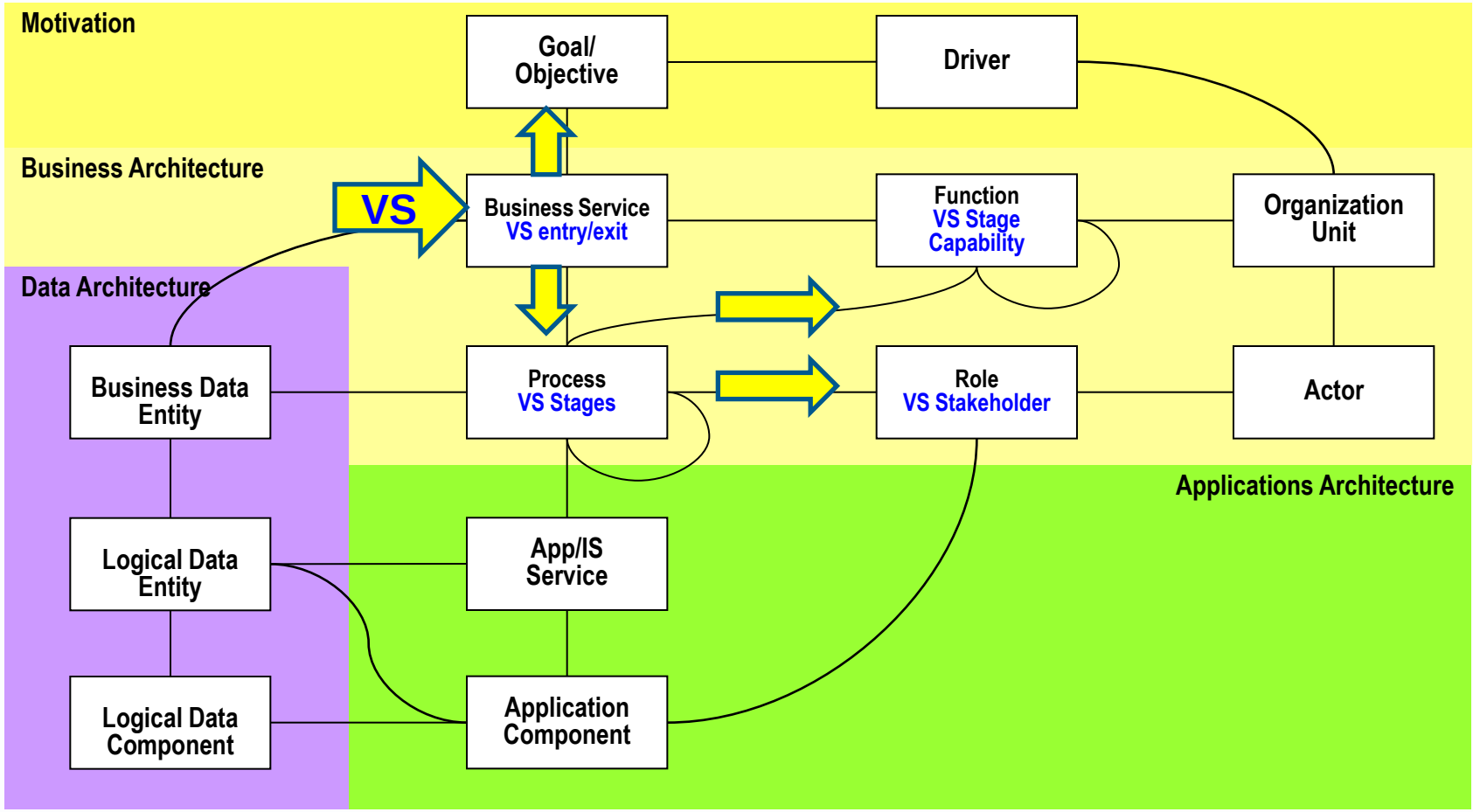
Example business scenario/value stream

Name	Capture order		
Goal, purpose, value added	As implied by the name above and exit facts below		
Roles	Customer Sales person		
Entry criteria	Trigger: Visit customer at scheduled time Input: Customer details Preconditions: Sales visit agreed and scheduled		
Exit criteria	Outputs or products: Signed order Post conditions: Order captured and recorded		
Activities	**Human activities**	**Applications**	**Capabilities**
	1 Initiate sales process with the customer		
	2 Discuss customer requirements		
	3 Work with customer to create a product configuration	Product Configurator	**Sales & Marketing**
	4 Verify that the desired configuration can be delivered	Inventory Scheduling	**Supply** **Delivery**
	5 Determine price of requested configuration	Pricing	**Sales & Marketing**
	6 Confirm customer desire to purchase		
	7 Place an order	Order entry	**Sales & Marketing**
	8 Capture customer signature	Order entry	**Sales & Marketing**
Non-functional qualities	Duration: 1 hour Throughput: 2 per day per salesman Availability: Working hours		

App/Capability matrix as a diagram



Value Stream in the meta model



Part 10: Adapting TOGAF to use BA terminology

BA in TOGAF as it is: Functions and Scenarios

- Given an enterprise **function hierarchy** *along with* a Request for Architecture Work.
- In phase A, you target selected **functions** and **business scenarios.**

ADM Deliverable	Enterprise Continuum Enterprise Repository	Services & Building Blocks	Business domain entities	Applications domain entities	Data domain entities	Technology domain entities
EA/Strategic Architecture			Function and Organization Hierarchies	Application portfolio catalog	Business data entity catalog	
Architecture Req'ments Specification	Req'ments & Context Req'ments Repository	Business & Application Service Contracts	Business Services Business Scenarios	App/IS Services		
Architecture Definition Document	Architecture Continuum & Repository	Architecture Building Blocks	Roles			
Architecture Road Map	Solutions Continuum & Repository	Solution Building Blocks				
Architecture Change Requests	Deployed Solutions		Identity Management	IT Configuration Management (CMDB)		
			Business & IT Operations			

TOGAF adapted to BA terms: Capabilities and Value Streams

- Given an enterprise-wide **capability map** *along with* a Request for Architecture Work.
- In phase A, you target selected **capabilities** and **value streams.**

ADM Deliverable	Enterprise Continuum Enterprise Repository	Services & Building Blocks	Business domain entities	Applications domain entities	Data domain entities	Technology domain entities
EA/Strategic Architecture			Business Capability and Organization Hierarchies	Application portfolio catalog	Business data entity catalog	
Architecture Req'ments Specification	Req'ments & Context Req'ments Repository	Business & Application Service Contracts	Business Services Value Streams	App/IS Services		
Architecture Definition Document	Architecture Continuum & Repository	Architecture Building Blocks	Roles			
Architecture Road Map	Solutions Continuum & Repository	Solution Building Blocks				
Architecture Change Requests	Deployed Solutions		Identity Management	IT Configuration Management (CMDB)		
			Business & IT Operations			

Recap

- This slide show presents TOGAF in a coherent and consistent way.
- And relates it to other architecture standards and sources

  - Part 1: Harmonising TOGAF with itself
  - Part 2: Harmonising TOGAF with data architecture
  - Part 3: Harmonising TOGAF with SOA
  - Part 4: Harmonising TOGAF with Zachman
  - Part 5: Harmonising ArchiMate with TOGAF
  - Part 6: Harmonising TOGAF with ArchiMate
  - Part 7: Radically changing TOGAF to fit ArchiMate?
  - Part 8: Harmonising TOGAF with Business Capabilities
  - Part 9: Harmonising TOGAF with Value Streams
  - Part 10: Adapting TOGAF to use BA terminology

- An aim is to help people ensure piecemeal change requests do not undermine the coherent and consistent view presented herein.