Avancier

Service-Orientation
in TOGAF and ArchiMate

From Logical to Physical

You may show this slide show provided you also show your audience where they can find it at http://avancier.website

GrahamBerrisford@gmail.com

The system theory context for business system description

- *Social systems* are composed of actors who communicate and perform activities according to information received.

- *Business systems* are social systems that are formalised and describable as discrete event-driven activity systems.

- Premises
  - All behaviour is event-driven, or discrete.
  - All behaviour is performed by active structural elements
    - active objects in UML
    - actors, components and nodes in ArchiMate.

Models *typify* things

- Architects usually model *types* of things
- *Instances* appear in the run-time system – in the deployed solution

ArchiMate aspect	Descriptive type	Real individual	
Active structure	**Role**	Actor	has a state and relationships to other actors
Behaviour	**Process** type	Performance	runs from start to end according to business rules
Behaviour	**Event** type	Occurrence	triggers a process performance
Passive structure	**Data** type	Data structure / item	encodes specific meaningful information (may be created, moved, changed or destroyed).

EA is not like machine or building architecture

- Mechanical engineers model systems that work on matter and energy.
- EAs model business processes that work on ***information***
  - And that information models the business itself.

- Building architects model passive structures.
- EAs model discrete-event driven ***activity systems***
  - where the primary requirement is for ***regular behaviours.***

Behaviour/structure is a time/space distinction

- System theory

System	Behaviours in time	Structures in space
External view	Events and results	I/O boundary
Internal view	Regular behaviours	Active components

- Email system

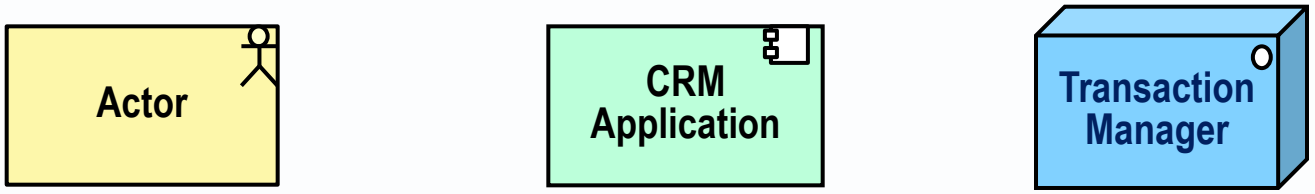
Email system	Behaviours in time	Structures in space
External view	Send email, Receive email	Human interface, API
Internal view	(invisible)	Email application

- ArchiMate terms

ArchiMate	Behaviours in time	Structures in space
External view	Services	Interfaces
Internal view	Processes	Actors/Components

Three premises

1. All behaviours in a system are performed by active components that occupy space and must be addressable.



2. All regular behaviours in a business system are triggered by discrete events, and run over time.

3. All system descriptions are abstract.
   - Using composition, generalisation, idealisation, delegation etc.

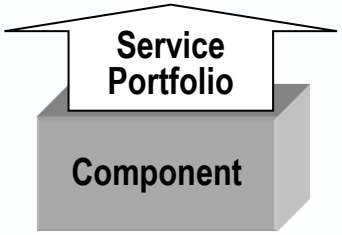
Service-oriented specification of components

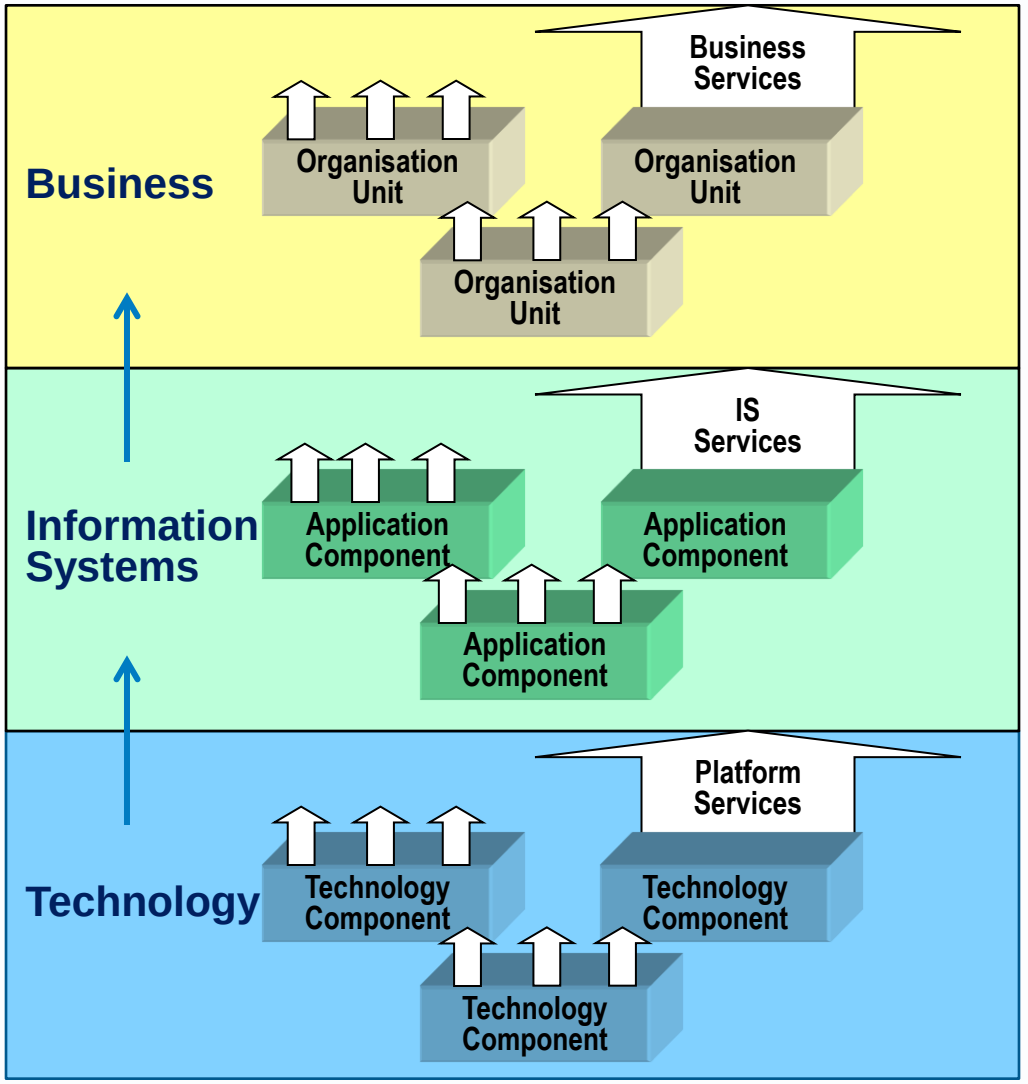
Avancier

1. **Service-oriented specification of components**
2. Realising logical by physical
3. Mapping ArchiMate to TOGAF
4. Things to beware of

TOGAF Architecture Domains and Building Blocks

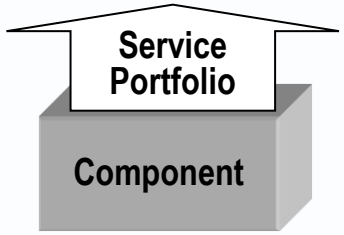
- Such a 3-layer view is commonplace in architecture frameworks



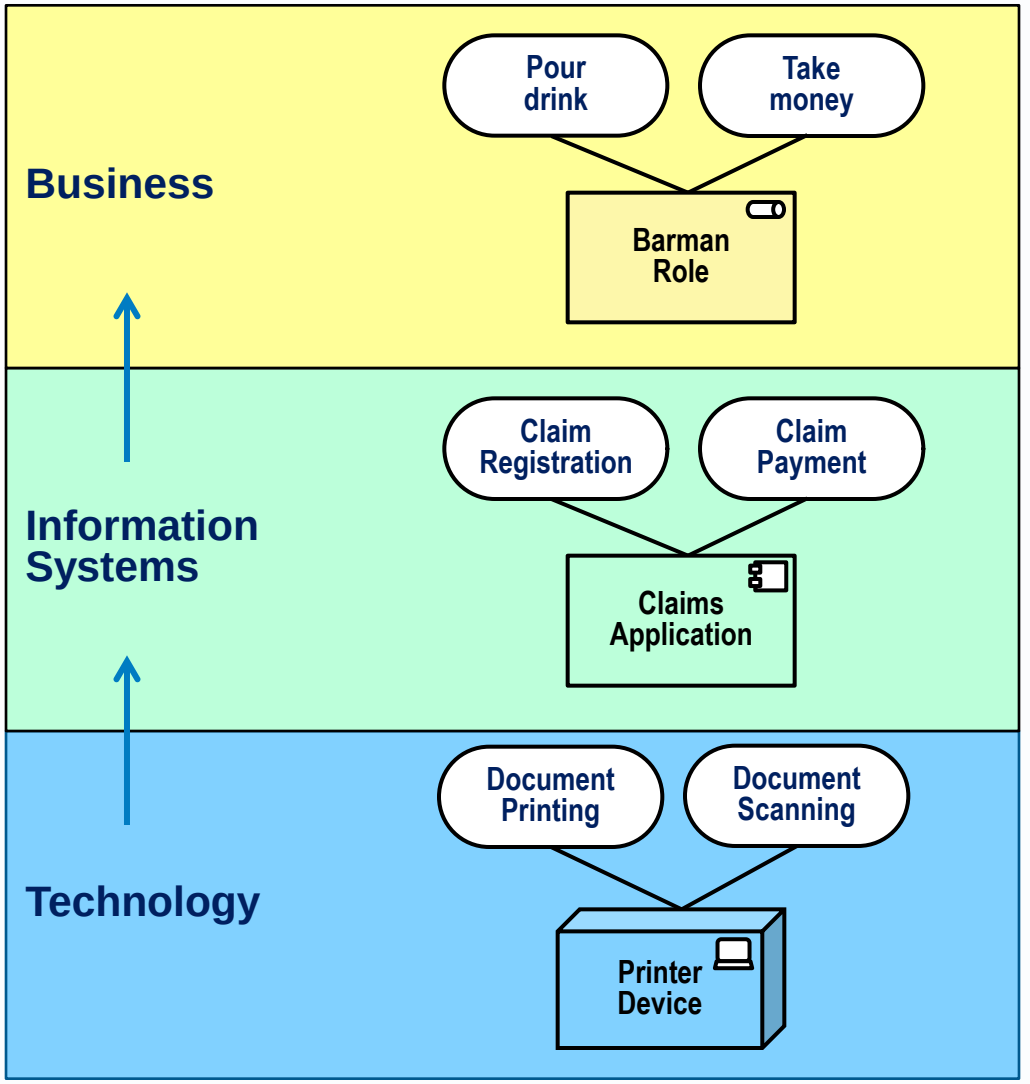


TOGAF applies CBD and SOA principles

- Components are building blocks
- Each component is defined by the service portfolio it provides.



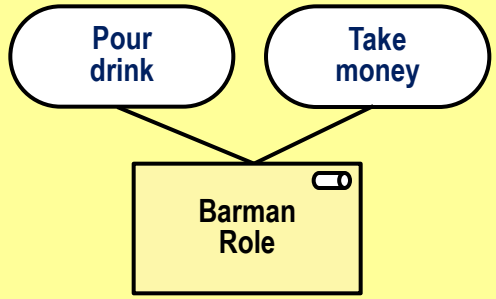
- (Though note that a coarse-grained service may require several fine-grained components)



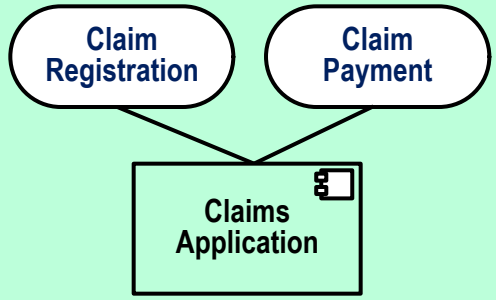
ArchiMate applies CBD and SOA principles

Avancier

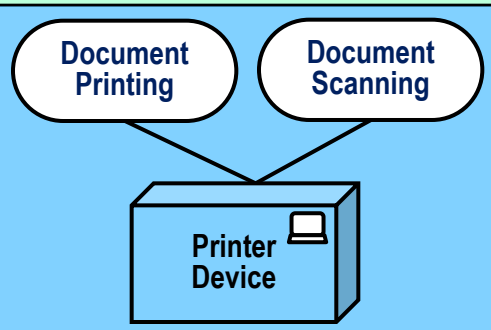
Business layer
"offers products and **services** to external customers realized in the organization by business processes performed by business **actors**."



Applications layer
"supports the business layer with **application services** realized by (software) **applications.**"



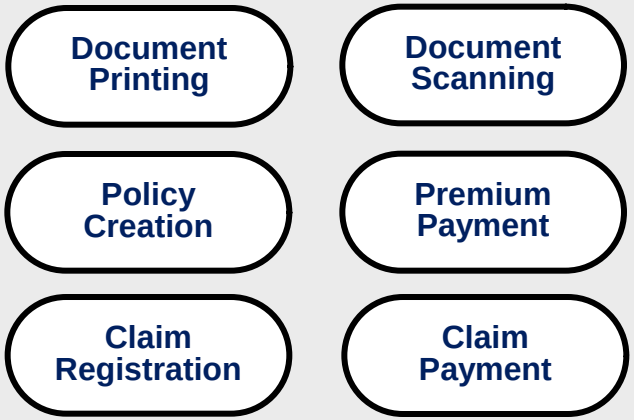
Technology layer
"offers **infrastructure services** needed to run applications realized by computer and communication **hardware and system software**"



“Service” in ArchiMate and in TOGAF

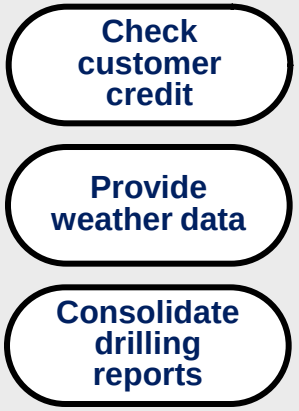
Avancier

- “a **unit of functionality** that a system exposes to its environment,
- hides internal operations,
- provides a value,
- accessible through interfaces.”



- Examples from ArchiMate

- “an **element of behaviour** that
- provides specific functionality in response to requests from actors or other services”
- “a logical representation of a repeatable business activity,
- has a specified outcome,
- is self-contained,
- is a ‘‘black box’’ to its consumers.”

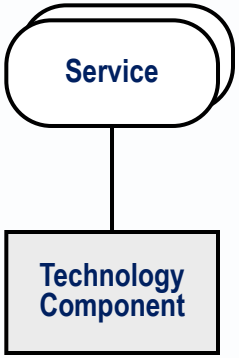


- Examples from TOGAF

The Open Group: services as the requirements for systems

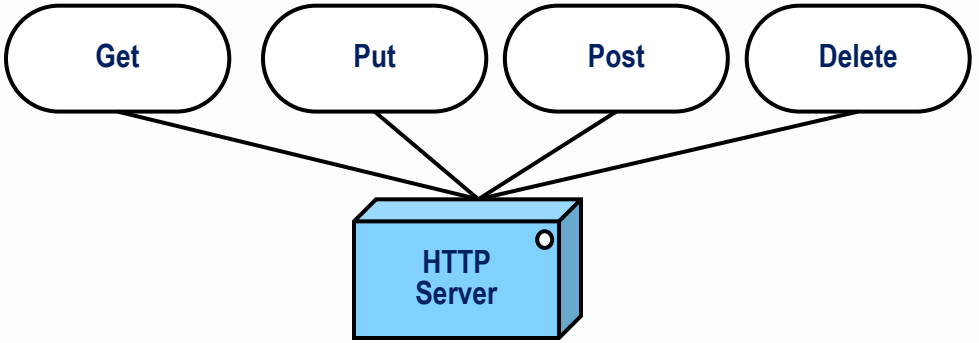
TOG was created to standardise systems through the **open** development and publication of **vendor and technology-neutral** (logical) specifications.

Principle: open standards (e.g. the Unix specification) define a system by itemising the *logical services* to be provided by its *physical components*.



For each building block, build up a service description portfolio as a set of non-conflicting services."

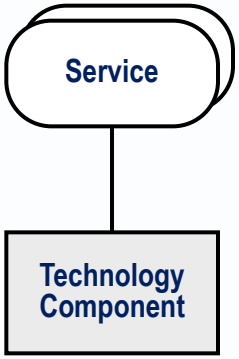
TOGAF 9.1 Phase D 12.4.1.6



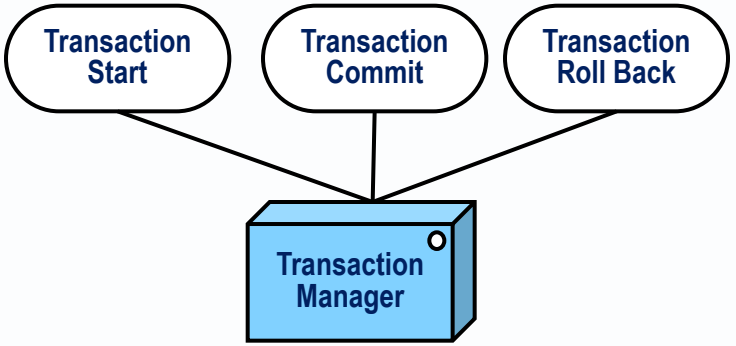
TOGAF 1 to 7 were based on a Technical Reference Model (TRM)

A TRM defines an enterprise's *complete infrastructure technology estate* by cataloguing all services it offers to business applications.

A subset of the TRM (a service portfolio) is assigned to a logical technology components.

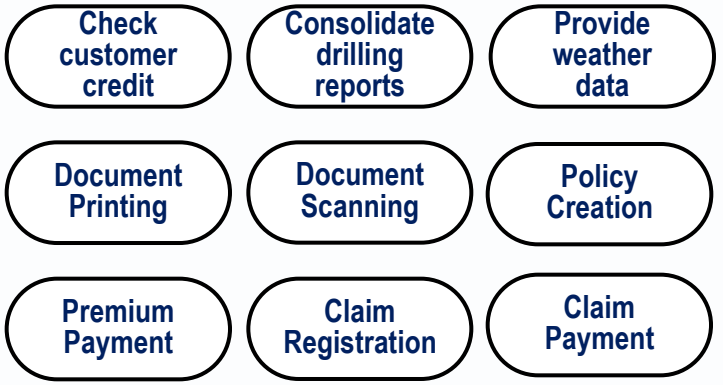


"The services portfolios are combinations of basic services from the service categories in the TRM, that do not conflict.



So what is a Service?

- A discrete operation requestable by a client:
  - Build a house, Fill a pothole
  - Haircut, Shampoo, Manicure
  - Train seat booking
  - Get, put, post or delete operation (HTTP)
  - Start, commit or rollback a transaction



A *required service* may be defined in a service contract without regard to the internal workings of any components or processes that *provide the service.*

Service contract

Signature: name, inputs and outputs.

Functional rules: preconditions and post conditions

Non-functional characteristics: inc. performance and commercial conditions.

"For the external users, only this external functionality, together with non-functional aspects such as the quality of service, costs etc., are relevant." ArchiMate.

Service Contract NOT = Service Level Agreement.
A SLA specifies a B-to-B Interface, and usually covers many discrete Services

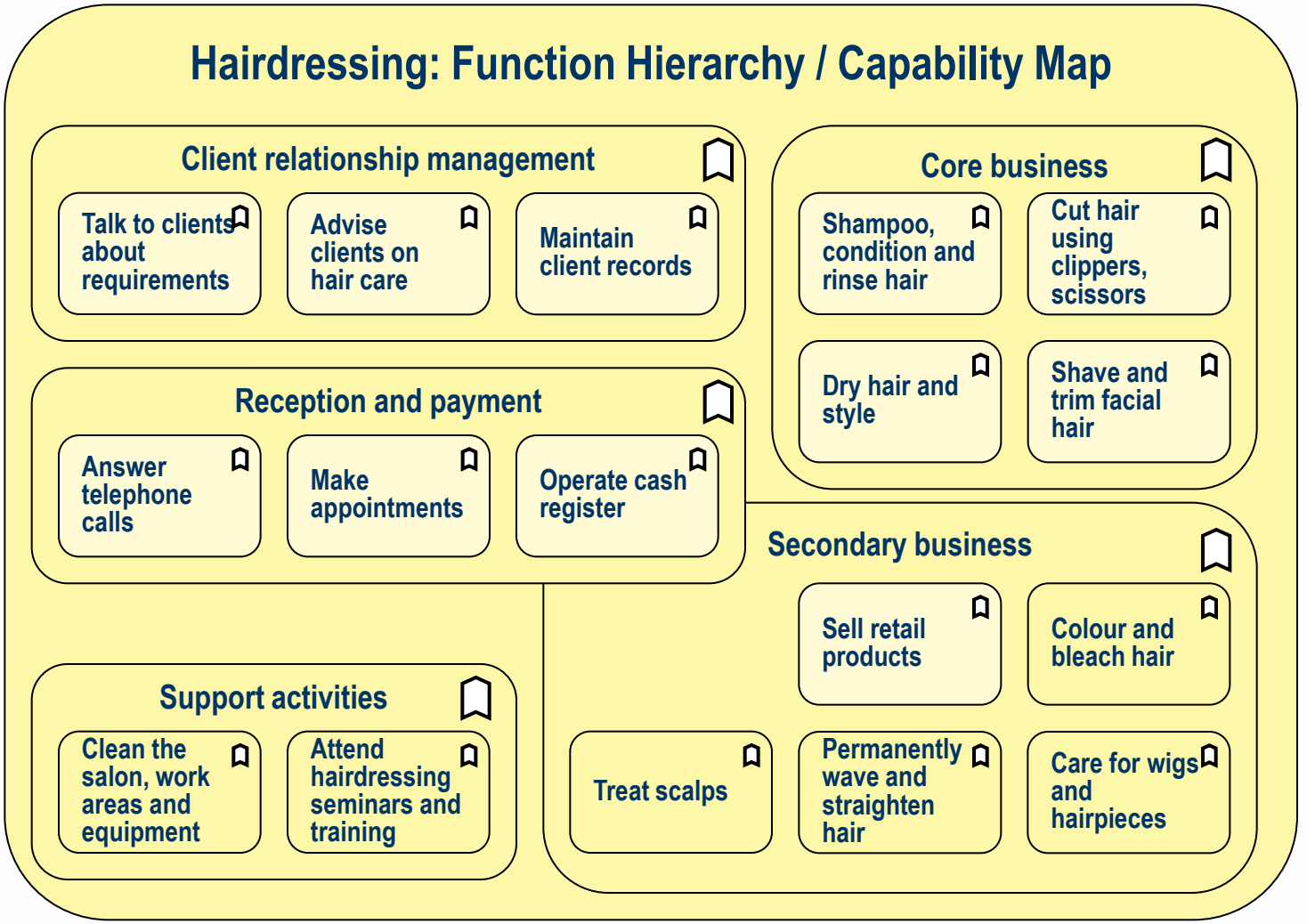
Q) What is the right level of granularity?

Avancier

- You describe component, process and service entities at whatever level of composition or decomposition you choose.
- Services are what can be requested of the system, capability or function you are defining. The granularity can be
  - coarse-grained (e.g. build me a house) or
  - fine-grained (get file).
- Services are external to the component (e.g. CRM application) whose boundary you are defining through service definition. But also internal to any wider system, function or capability of interest.
- See footnote for more on this theme.

A structural view of activities

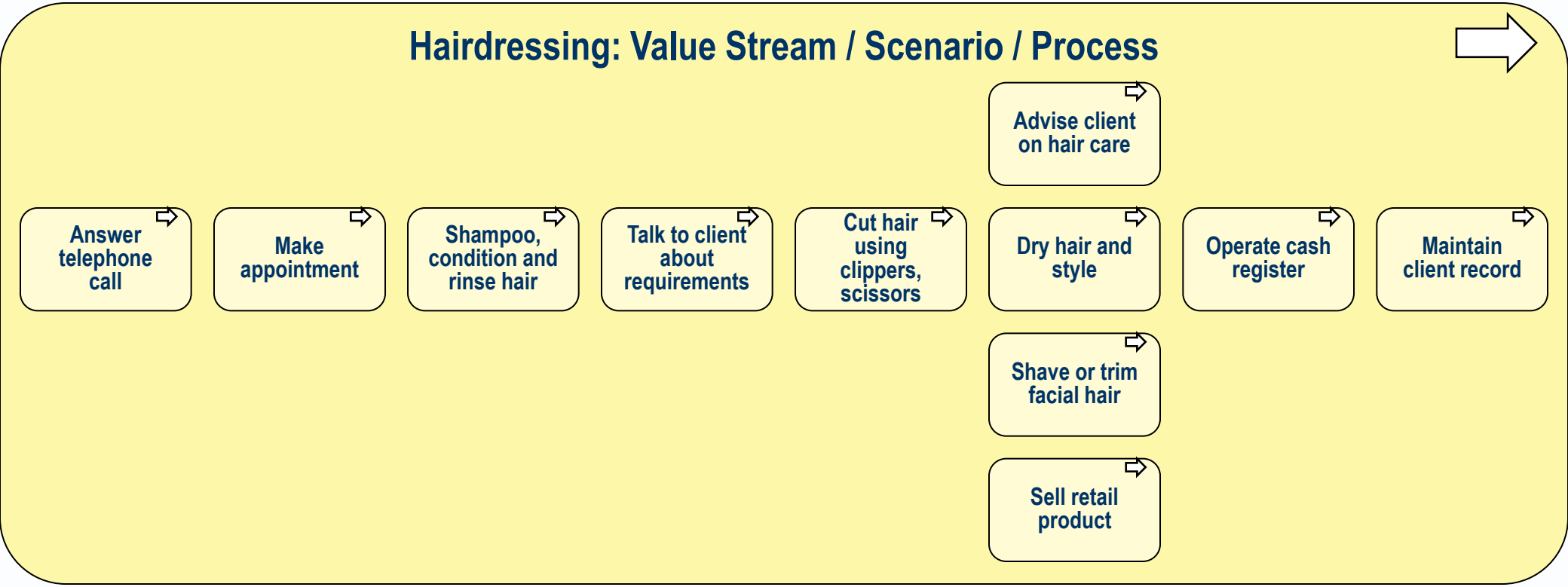
- Each node in the structure is definable in terms of services offered and/or subordinate functions/activities
- Often decomposed to 3 or 4 levels



A behavioural view of the *same* activities

Avancier

- Triggered by an event, ends in result

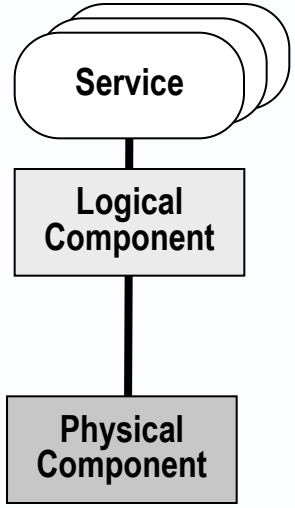


- Some plurals become singular; some “ands” become “ors”

Realising Logical by Physical

Avancier

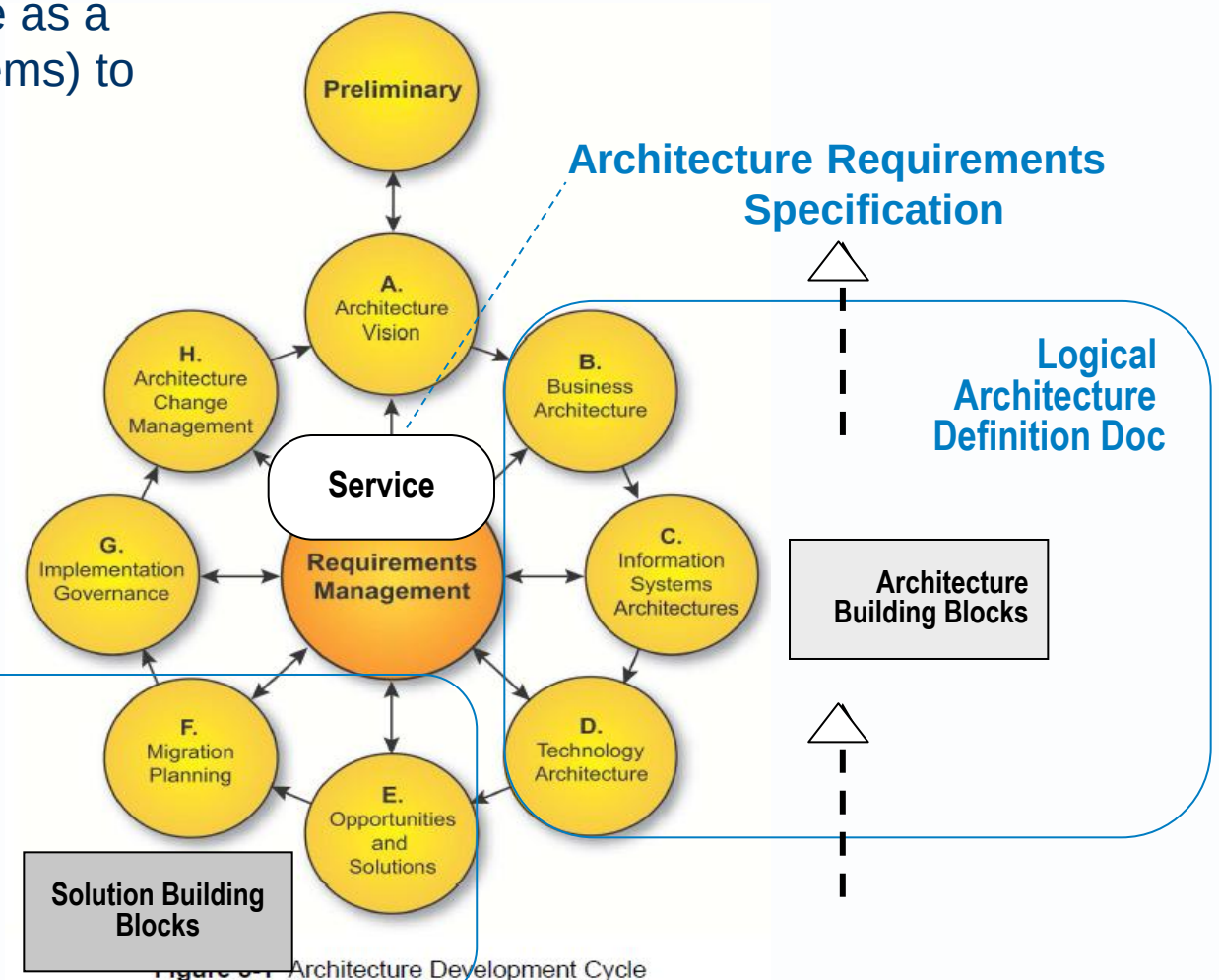
1. Service-oriented specification of components
2. **Realising logical by physical**
3. Mapping ArchiMate to TOGAF
4. Things to beware of



Logical to physical in TOGAF

Avancier

TOGAF regards the enterprise as a system (or system of systems) to be described *abstractly.*

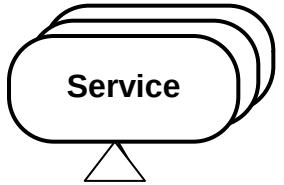
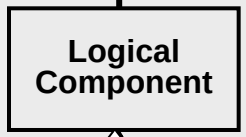
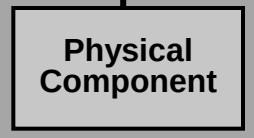


Physical Solution Options

"The physical elements in an enterprise architecture may still be *considerably abstracted from Solution Architecture*"

In TOGAF, a component is a kind of “Building Block”

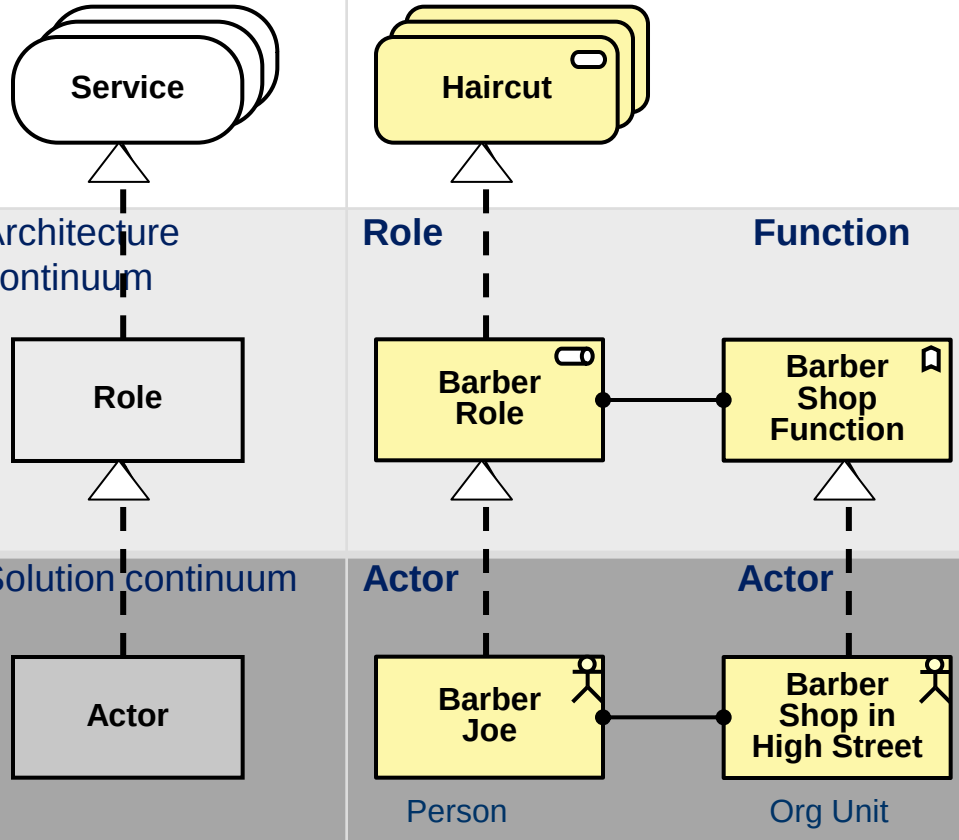
Avancier

TOGAF		ArchiMate	
Service A requestable activity/operation/process, definable in a contract, and ideally found in only one ABB service portfolio.	Requirements 	**Service**	
Logical Component An "architecture building block" (ABB) An ideal or potential component Defined by services offered (or activities performed) Vendor and technology neutral.	Architecture continuum 	**Role, Function**	
Physical Component A “solution building block" (SBB) A real component that can perform activities and implements logical component(s) Vendor or technology specific.	Solution continuum 	**Actor, Component, Node**	

E.g. Business architecture

Avancier

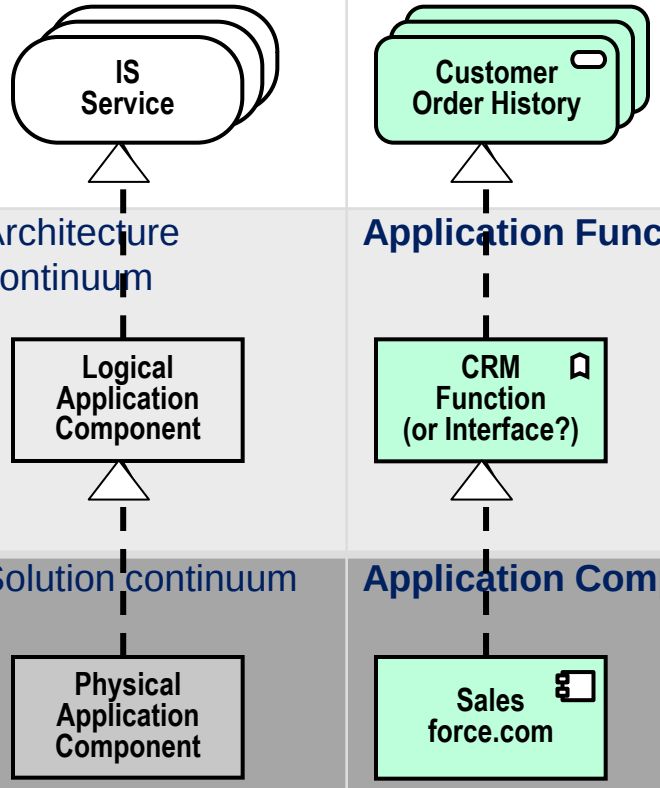
TOGAF business architecture		ArchiMate
Business Service What does a customer want?	Requirements	**Business service**
Role or Function What role or function do we need to provide the service?	Architecture continuum	**Role** **Function**
Actor or Organisation Unit What actor or organisation unit can we acquire to play the role or perform the function?	Solution continuum	**Actor** **Actor**



E.g. Applications Architecture

Avancier

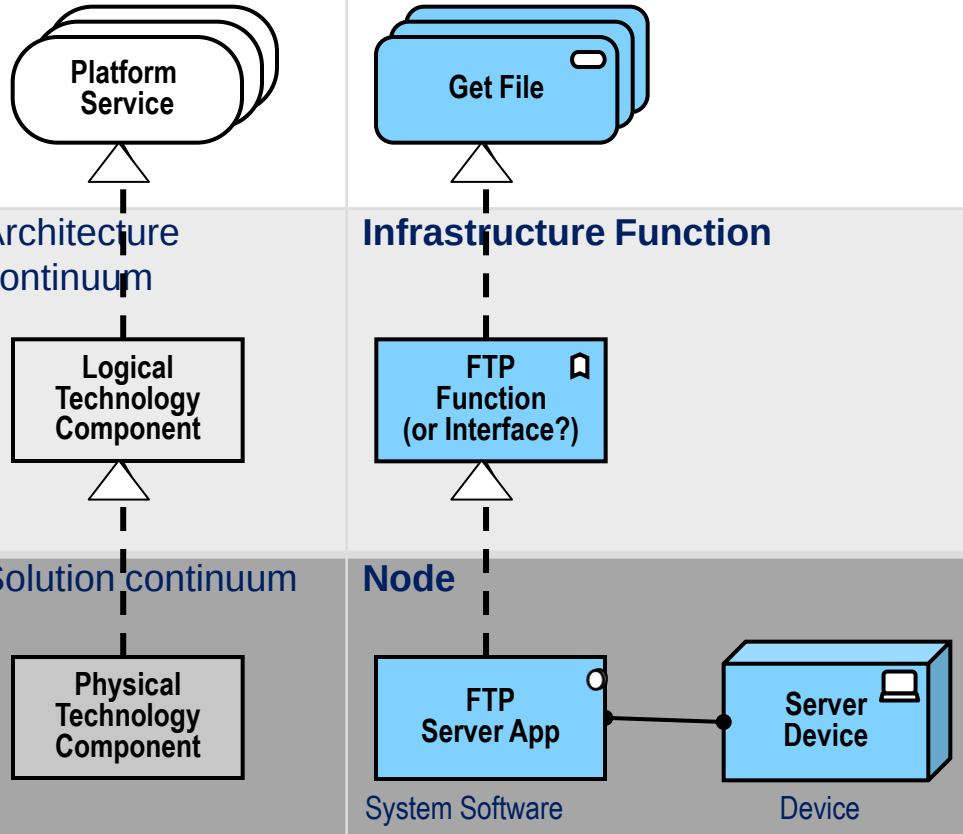
TOGAF information systems architecture		ArchiMate
IS Service Your sales Organisation unit require 20 IS Services (use cases) from a logical component.	Requirements	**Application Service**
Logical Application Component You call it a customer relationship management system, which could be realised by a human activity system or by any of several COTS (“packaged”) computer applications.	Architecture continuum	**Application Function**
Physical Application Component You choose a specific physical application because it offers 18 of the 20 of the required services. It offers 5 other services you never thought to ask for, which are “opportunities”.	Solution continuum	**Application Component**



E.g. Infrastructure technology

Avancier

TOGAF technology architecture		ArchiMate
Platform Service Ideally, selected from the enterprise TRM	Requirements	**Infrastructure Service**
Logical Technology Component Defined by "service portfolio“ the ABB is to provide. E.g. the IETF standard FTP interface.	Architecture continuum	**Infrastructure Function**
Physical Technology Component The (SBB) you hire, buy or build to realise the ABB. E.g. the particular FTP server you deploy on your device(s).	Solution continuum	**Node**



TOGAF text and references

Avancier

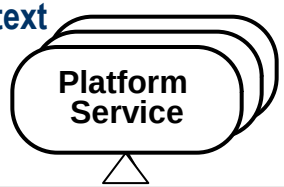
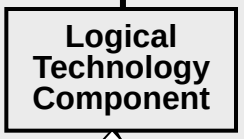
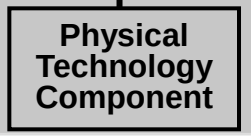
Enterprise Continuum		
Requirements & context 		
Architecture Continuum: a repository of architectural elements…" 	**"Logical:** An implementation-independent definition of the architecture, often grouping related physical entities according to their purpose and structure."	**"Architecture Building Block (ABB):** a constituent of the architecture model." "Architecture Building Blocks: Architecture documentation and models…"
Solutions Continuum: contains implementations of… definitions in the Architecture Continuum." 	**"Physical:** A description of a real-world entity. Physical elements in an enterprise architecture may still be considerably abstracted from Solution Architecture, design, or implementation views."	**"Solution Building Block (SBB) :** a candidate physical solution for an Architecture Building Block (ABB)" "Solution Building Blocks: Implementation-specific building blocks…"

Figure 2-3 Enterprise Continuum
Figure 6-2 Management Frameworks to Co-ordinate with TOGAF
Figure 6-3 Interoperability and Relationships between Management Frameworks
Figure 28-2 Consolidated Gaps, Solutions, and Dependencies Matrix
Figure 40-1 Summary Classification Model for Architecture Landscapes AND Figure 40-2 Summary Classification Model for Solutions.

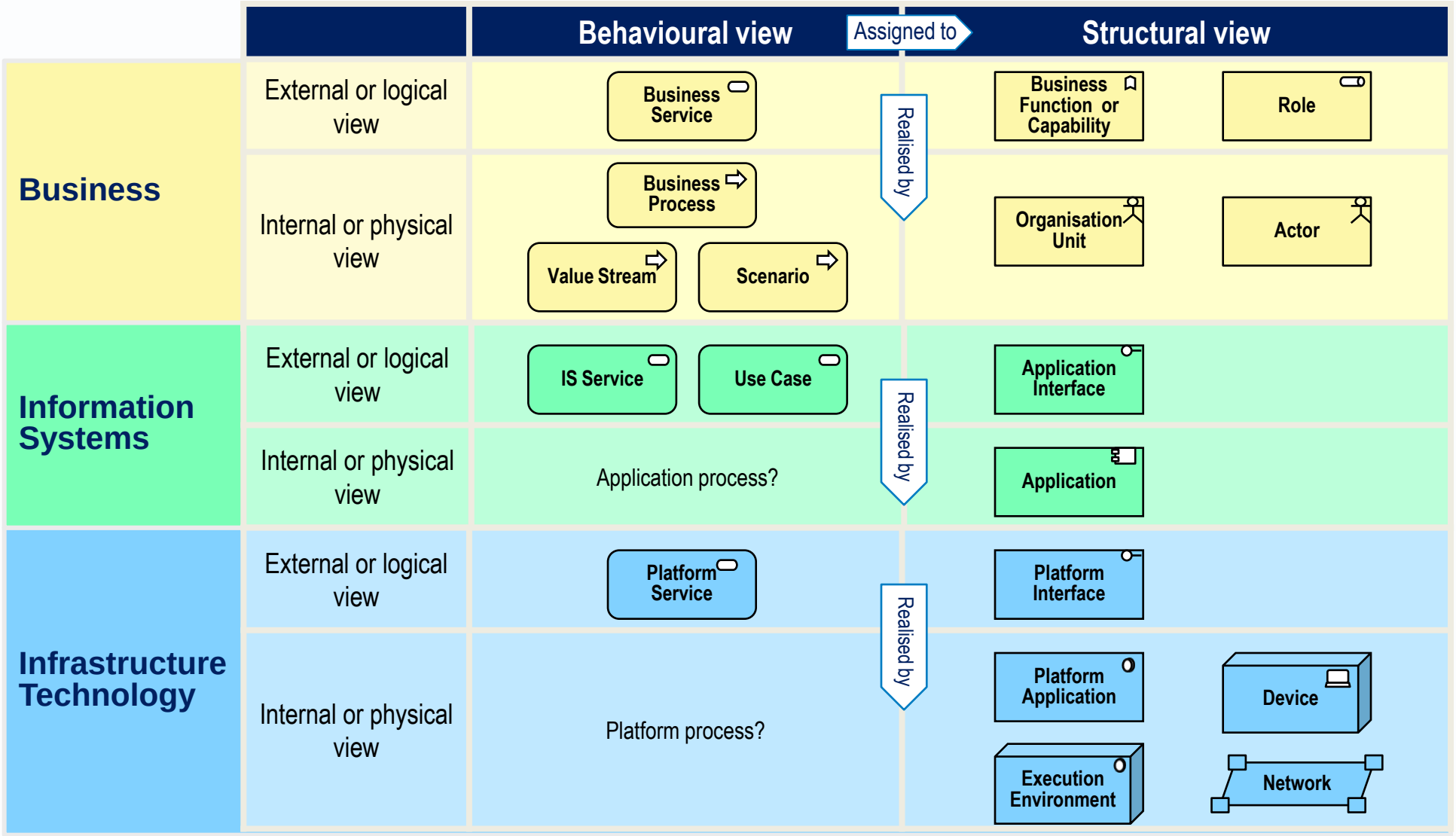
Mapping ArchiMate to TOGAF

Avancier

1. Service-oriented specification of components
2. Realising logical by physical
3. **Mapping ArchiMate to TOGAF**
4. Things to beware of

EA terms and concepts, drawn using ArchiMate symbols

Avancier



Mapping TOGAF to ArchiMate 1

Avancier

In the standard

Not in the standard

- Meaning gets lost because
- people use the terms “service” and “function” loosely and ambiguously,
- there are more concepts than people know how to use

TOGAF	External Behaviour	Internal Behaviour	Logical structure of services and activities		Physical Structure
Business	Business Service	Process Elementary Process	Function Role	SLA Job Spec	Org Unit Actor
Information Systems	IS Service	IS Process	Log App Component	App Interface	Phys App Component
Technology	Platform Service	Platform Process	Log Tech Component	Technology Interface	Phys Tech Component

ArchiMate	External Behaviour	Internal Behaviour	Logical structure of services and activities	External Structure	Internal Structure
Business	Business Service	Business Process Elementary Process	Business Function Role	Business Interface Job Spec	Org Unit Actor
Applications	App Service	App Process	App Function	App Interface	App Component
Technology	Infrastruct. Service	Infrastruct. Process	Infrastruct. Function	Infrastruct. Interface	Node

Mapping TOGAF to ArchiMate 2

- Notice how TOGAF divides the structural view into logical and physical views

TOGAF	Passive Structure	Required Behaviour	Logical Structure	Physical Structure
Business		Business Service Process	Function Role	Org Unit Actor
Information Systems	Data Entity	IS Service	Log App Component	Phys App Component
Technology		Platform Service	Log Tech Component	Phys Tech Component

- Not used in this slide show

ArchiMate	Passive Structure	Required Behaviour	Active Structure	
Business		Business Service Process	Business Function Role	Actor
Applications	Data Object	App Service	App Interface	App Component
Technology		Infrastruct. Service	Infrastruct. Interface	Node

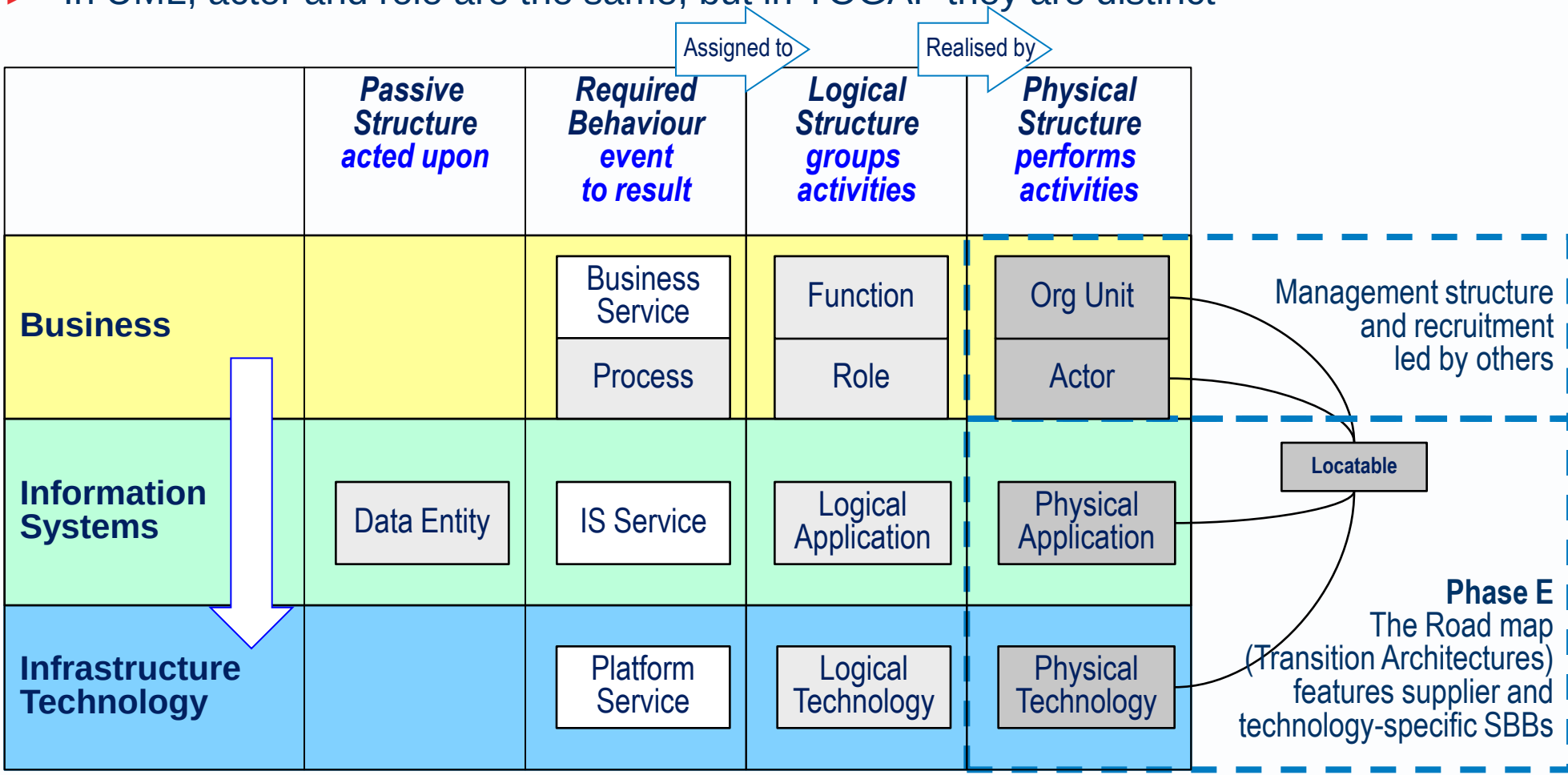
Business Interface

App Function

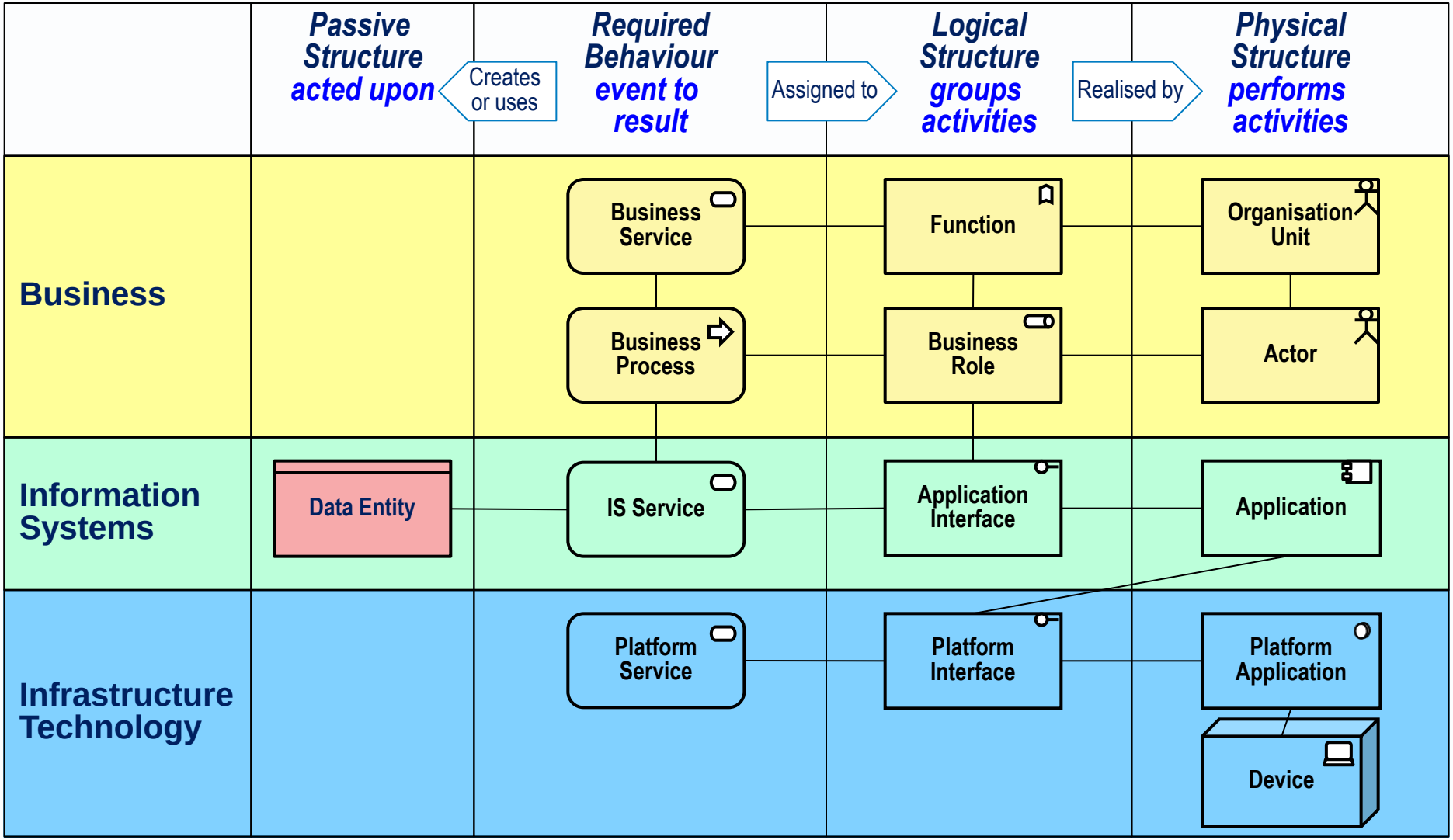
Infrastruct. Function

Mapping logical structure to physical structure in TOGAF

- In UML, actor and role are the same, but in TOGAF they are distinct

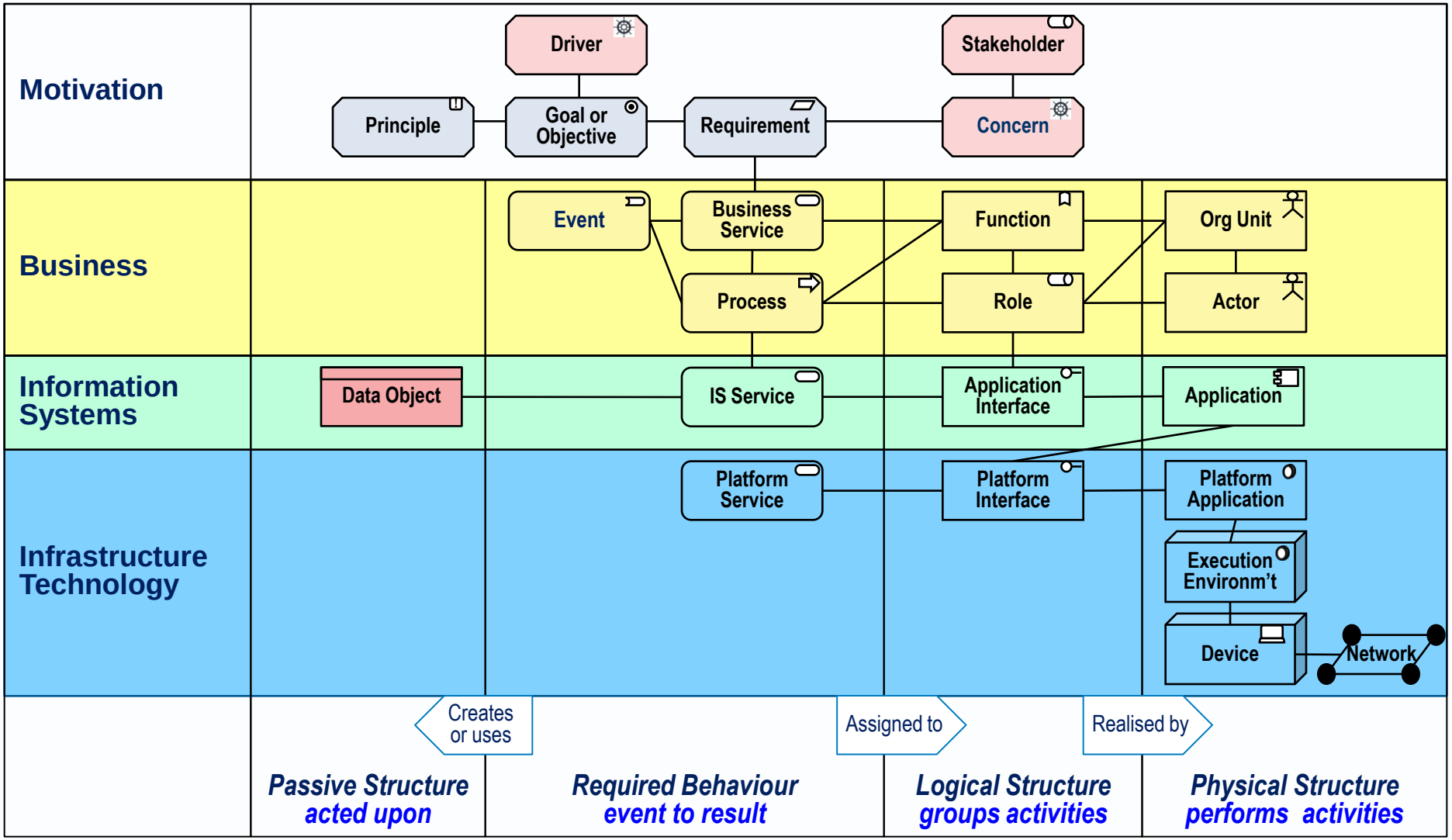


Avancier Methods core framework for EA with ArchiMate (level 1)



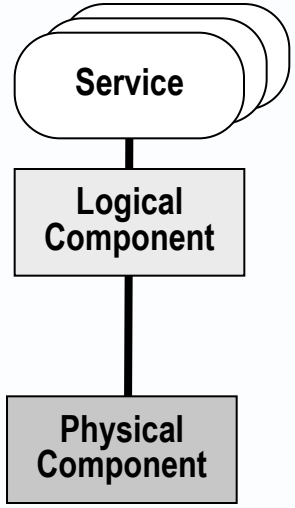
Avancier Methods core framework for EA with ArchiMate (level 2)

Avancier



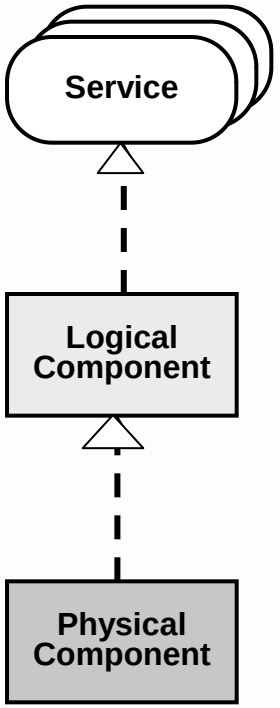
Things to beware of

1. Service-oriented specification of components
2. Realising logical by physical
3. Mapping ArchiMate to TOGAF
4. **Things to beware of**



Beware “logical” has at least four meanings out there

- Computation-independent
  - Conceptual, independent of computing
- External - encapsulated view
  - An interface or service portfolio
  - Regardless of internal content or workings
- Platform independent
  - Related to computing but independent of any specific vendor or technology solution
- Simple, elegant
  - Normalised and de-duplicated
  - Regardless of speed, throughput or other NFRs



Beware Service Contract not = SLA

Avancier

- “Services" are discrete (atomic events in the eyes of a client) and definable using a service contract

Service contract

Signature: name, inputs and outputs.

Functional rules: preconditions and post conditions

Non-functional characteristics: inc. performance and commercial conditions.

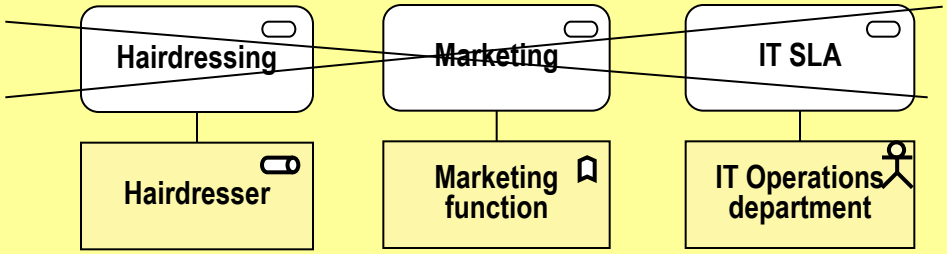
“For the external users, only this external functionality, together with non-functional aspects such as the quality of service, costs etc., are relevant.” ArchiMate.

A SLA specifies a B-to-B Interface, and usually covers many discrete Services

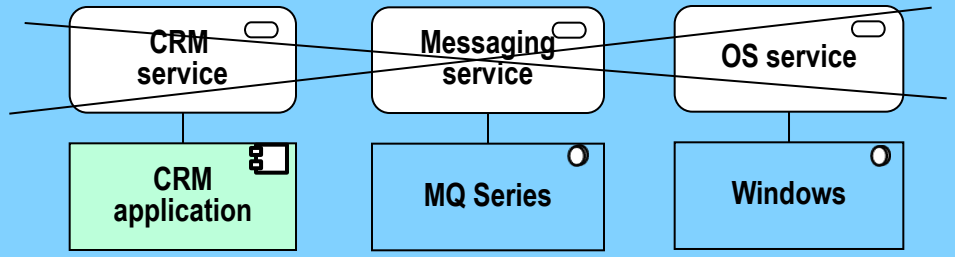
- (The discrete event-driven service is a valuable and necessary concept in the definition of systems, functions and capabilities. Unfortunately, core terms like “service” are used variously and loosely in practice. Asked to agree the definition of a word, a committee may relax the definition to the point of meaningless ambiguity. And so, valuable and necessary concepts get lost in the fog. Compromising definitions to suit all comers and all practices leads to ambiguity and incoherence. You have to bite the bullet and used a “controlled vocabulary.”

Beware “Service” is used to mean “Interface” “SLA” or “Availability”

To some business architects,
a service is an **SLA** or an **interface** to
all services of a role, function or actor



To some infrastructure architects,
a service is the **availability**
a component (its services) to users



- If you cluster discrete services , better the use the function symbol

Remember the two fundamental principles of TOG

- Principle 1: define open standards by itemising the logical services to be provided by physical components
  - Accordingly, TOGAF presumes 1 coarse-grained technology component will provide N finer-grained services, each selected from the TRM
- Principle 2: the "boundaryless information flow" vision embedded in the TOG mission statement.
  - Accordingly, TOGAF presumes 1 coarser-grained IS service can coordinate N finer-grained services from N distributed application components.
  - In the III-RM
    - An Information Consumer App requests a service from
    - A Brokering App, which requests N services from
    - N Information Provider Apps, which each access data on a particular data server.

Beware the impact of distributing business information

- 1 business information service request may require N Information Provider Apps to execute a database transaction on its own data server.
- But still, logically-speaking, the whole business information service is a "federated transaction“ you would like to treat as 1 atomic service

- What if the data servers are so distributed that a federated transaction cannot be treated (committed or rolled back) as 1 atomic service?

- Then the 1 federated transaction becomes a workflow or business process in the human activity system, orchestrating finer-grained services
- The business architect has to define exception cases and “compensating transactions” (a feature of CQRS)
- Since the design options have an impact on customers, employees and other stakeholders, it is important that business and application architects understand the implications of distributing business information.

Avancier Methods

- are useful with all architecture frameworks that share similar domains and entities

- http://avancier.website