Avancier Methods (AM)

Concepts

Architecture as Abstract Description of Systems

Architecture as abstraction

Avancier

- **“EA regards the enterprise as a system, or system of systems.”**

  TOGAF

- “IT systems are among the most complex systems ever built by humans, with millions of inter-connected working parts.”

  Roger Sessions

- “The goal of abstraction is to describe things more simply while retaining the information important to stakeholders.

  Class member Laurie Monk

  (after Einstein)

- **“Enterprise architecture is considerably abstracted from solution architecture”**

  TOGAF

- You cannot properly understand architecture until you understand abstraction.”

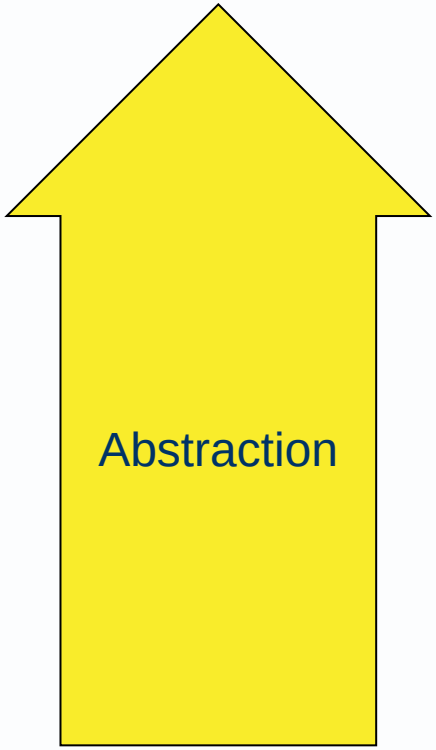
  Class member Bavo De Ridder

.

We all use abstraction every day

- **Animal intelligence** is forming, holding and using useful abstract models of the world the animal lives in
- **Business intelligence** is forming, holding and using useful abstract models of the world the business operates in
- **Architectural system descriptions** abstract from operational systems.
- The larger and more complex the operational system, the more **levels** of abstraction we need to describe them.

Levels of abstraction mapped to architect roles



- Enterprise architects
  - "EA is considerably abstracted from Solution Architecture, design, or implementation views." (TOGAF).
- Solution architects
  - describe systems at a more concrete level than EA, more abstract than builders
- Builders and technicians
  - describe systems at the bottom level of detail
- Operational systems
  - the run-time reality where processes are performed by actors/components

What kinds of abstraction are there?

- The larger and more complex the operational system, the more **levels** of abstraction we need to describe them.

- And in doing this, we entangle different **kinds** of abstraction – often without thinking about them.

Four kinds of abstraction used in higher level design

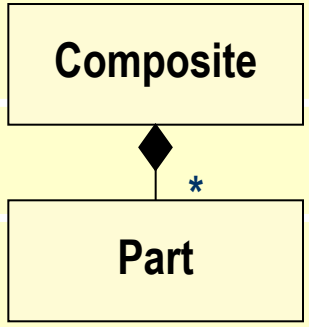
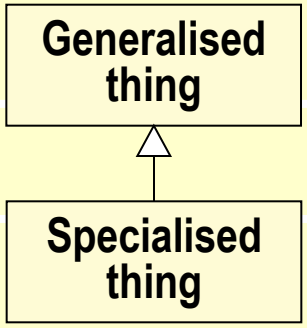
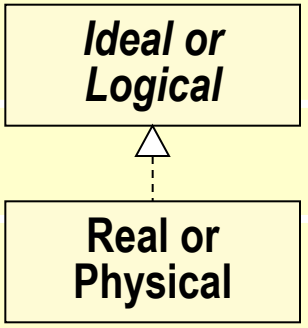
Avancier

Higher level design		Lower level design
Strategies and road maps	Longer time -> Shorter time	Shorter term sprints and deadlines
Broader goals, longer processes and coarser-grained subsystems	Composition -> Decomposition	Narrower requirements, shorter processes and finer-grained components
Standards, principles, patterns and reference models	Generalisation -> Specialisation	Application of standards, principles, patterns and reference models
Business needs and idealised system descriptions	Idealisation -> Realisation	Physical technology solutions
Encapsulation by definition of services in interfaces	External -> Internal	Realisation by internal roles and processes
Required processes	Behavioural -> Structural	Designed components

Three notations for abstraction

Avancier

- Notations used in UML and ArchiMate

	Composition packing smaller things inside bigger things	**Generalisation** removing differences between things	**Idealisation** removing differences between physical forms
			
	Elementary part	Uniquely configured	Physical Material
	Decomposition	**Specialisation**	**Realisation**

Four kinds of abstraction to be discussed

Avancier

Omission leaving out details	**Composition** packing smaller things inside bigger things	**Generalisation** removing differences between things	**Idealisation** removing differences between physical forms
Vacuous	**Coarse-grained composite**	**Universal**	**Conceptual Model**
Sketchy	**Mid-grained composite**	**Fairly generic**	**Logical Model**
Elaborate	**Fine-grained composite**	**Fairly specific**	**Physical Model**
Complete	Elementary part	Uniquely configured	Physical Material
Elaboration	**Decomposition**	**Specialisation**	**Realisation**

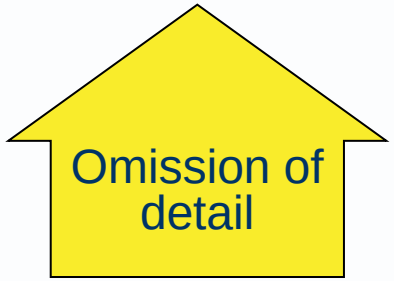
Avancier Methods (AM)

Concepts

Omission: The most general kind of abstraction

To create a management overviews you can…

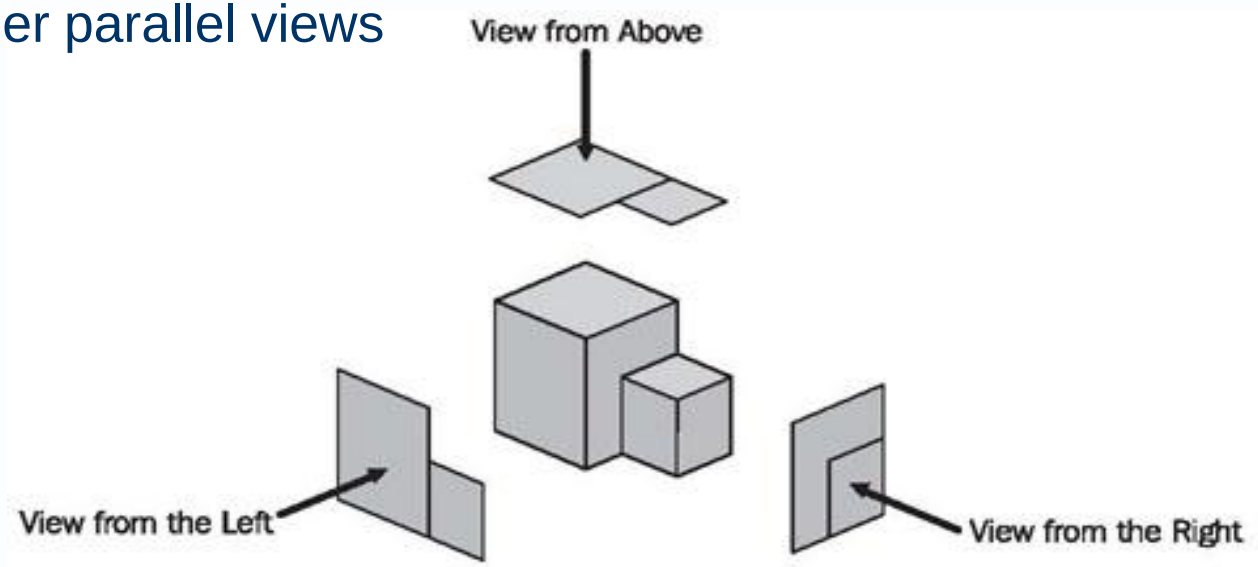
- **Omit** details in a lower level description



Omission		
Vacuous	**Doc name**	**Solution name**
Sketchy	**Abstract**	**Solution Vision**
Elaborate	**Distilled text**	**Solution Outline**
Complete	**Full text**	**Solution to Build**
Elaboration		

To suit different stakeholders you can…

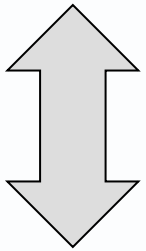
- **Omit** what is shown in other parallel views
  - Structural
  - Plumbing
  - Electrical



- Views may clash
  - A physical view of a book
    - The binding contains pages which contain printed shapes
  - A logical view of a book
    - The book contains chapters, which contain paragraphs, which contain sentences, which contain words, which contain letters.

Omission: definition

- **Omission:** Removal of details or perspectives from a description.
- Creates a digest or summary of a more elaboration description.
- Allows viewers to focus on what matters to them.
- Also known as Aristotelian idealization



- **Elaboration:** Addition of details or perspectives to a description.
- Makes the description more useful to builders

Avancier Methods (AM)

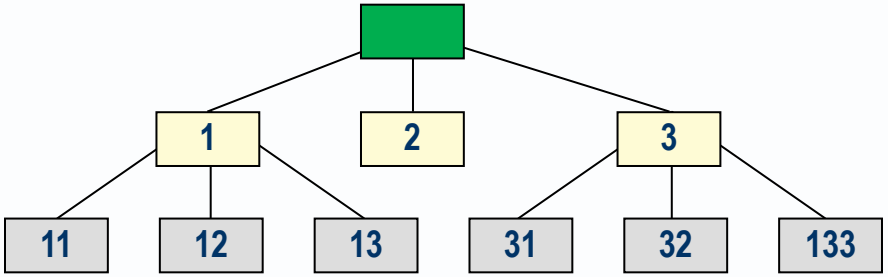
Concepts

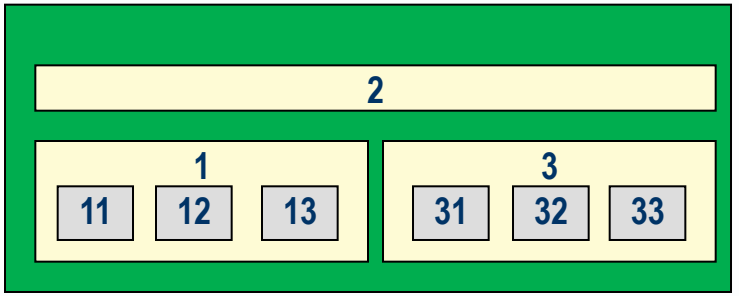
Abstraction by **Composition**

To direct a large social group

- You need a hierarchy

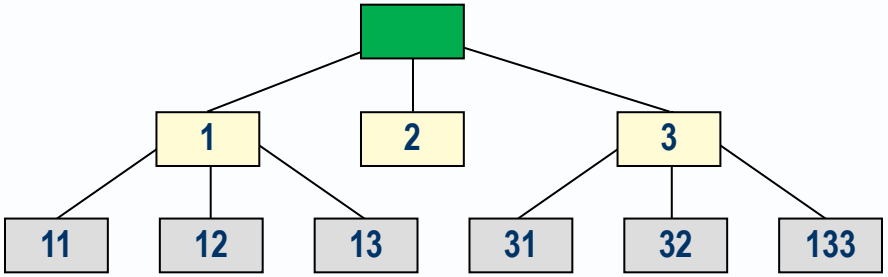
- Hierarchies can be drawn as a tree or nested boxes

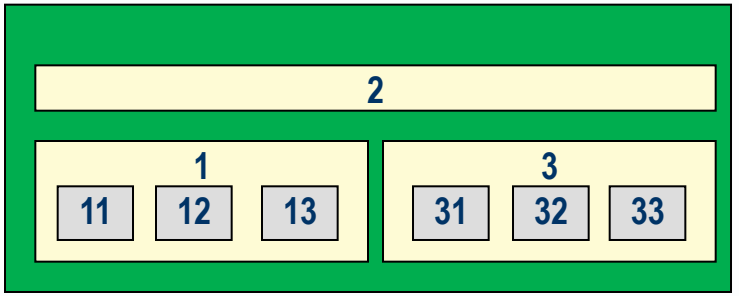




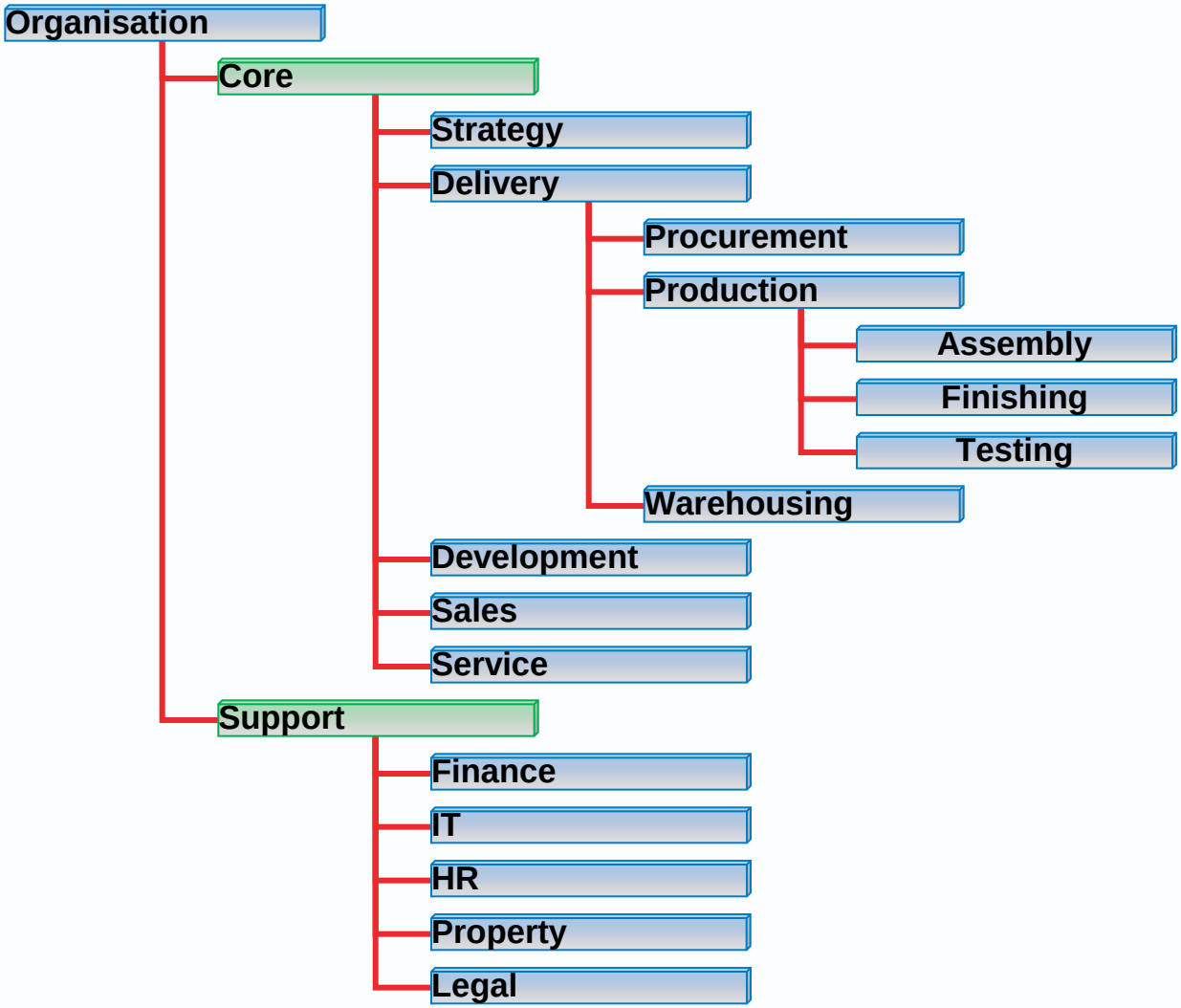
Architects use composition and decomposition in describing

1. Human organisation / management structures
2. A cascade of goals and objectives
3. Geographical organisation
4. Decomposition of system behaviour into process steps
5. Decomposition of system structures into components





1. Human organisation / management structures



Aka Capability Map

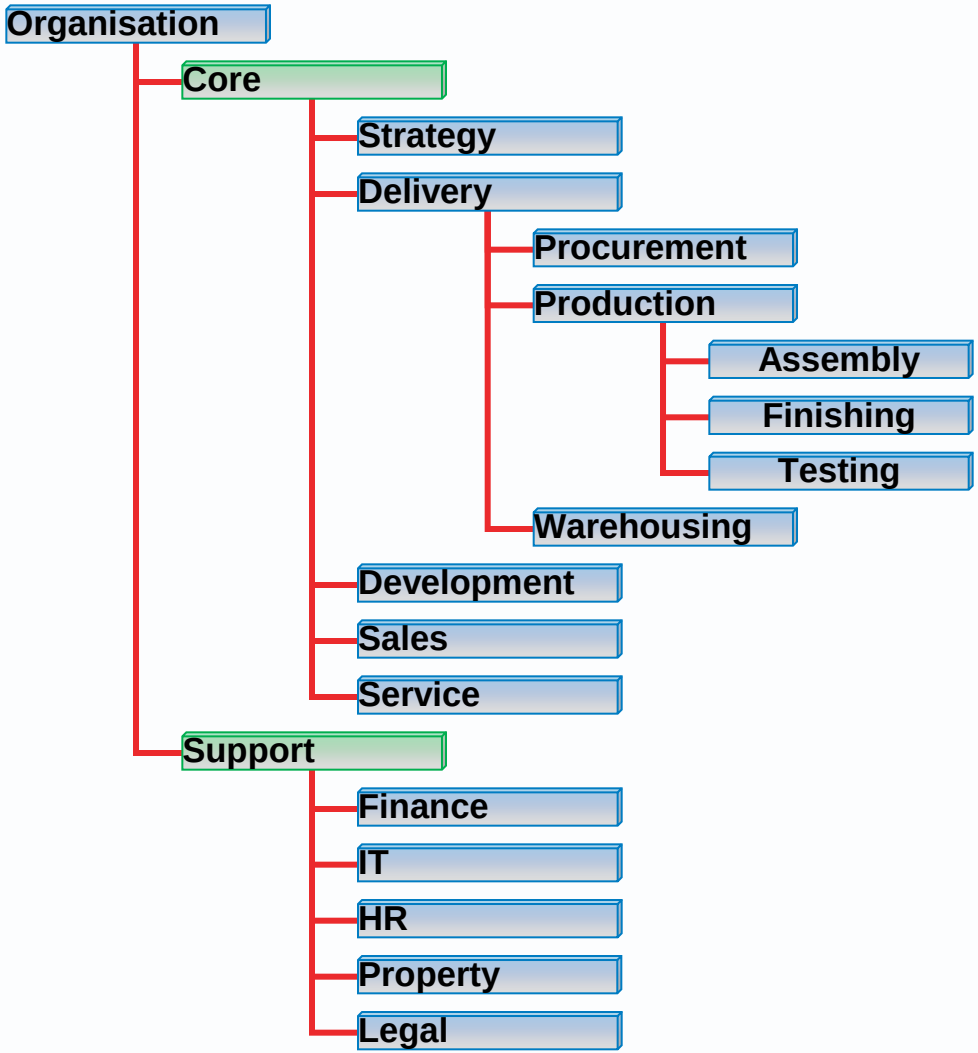
A *logical* functional decomposition for a bank

Avancier

The BIAN Service Landscape V2.5

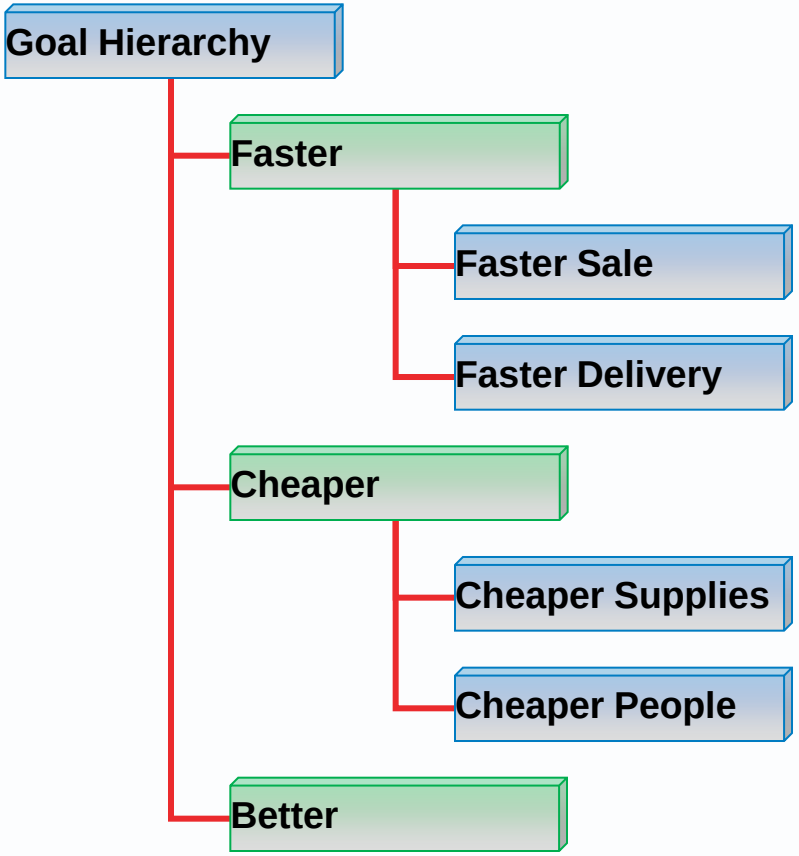
Logical decomposition usually stops at the 3rd or 4th level

- A structure with four levels is as deep as most system architects can manage.
- It is difficult to maintain the integrity of a structure with more levels (or more than thousand nodes).
- Difficult to avoid or manage duplication of lower level entries

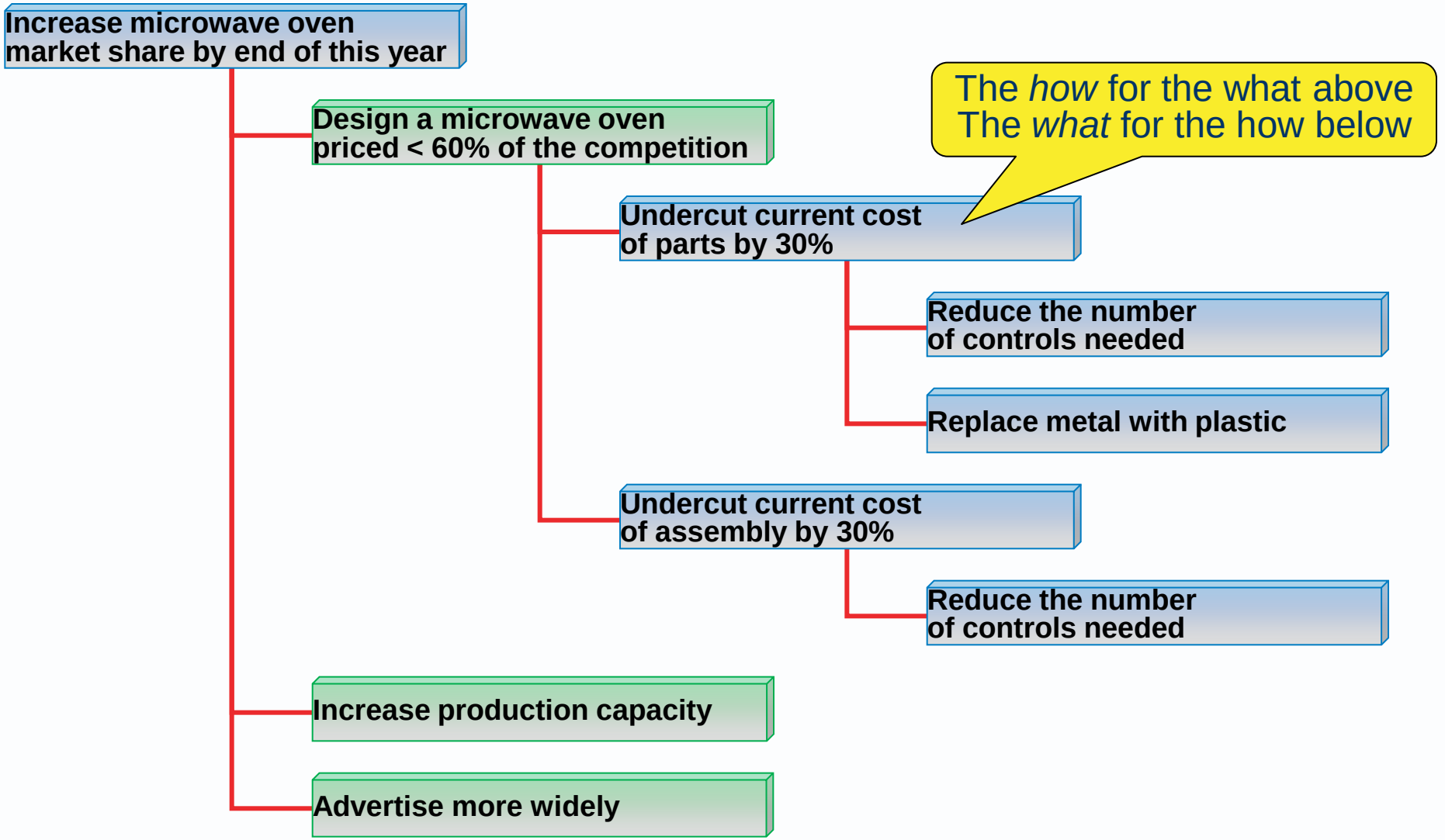


2. A cascade of goals and objectives

- Decomposition to define actionable objectives and assign them to actors

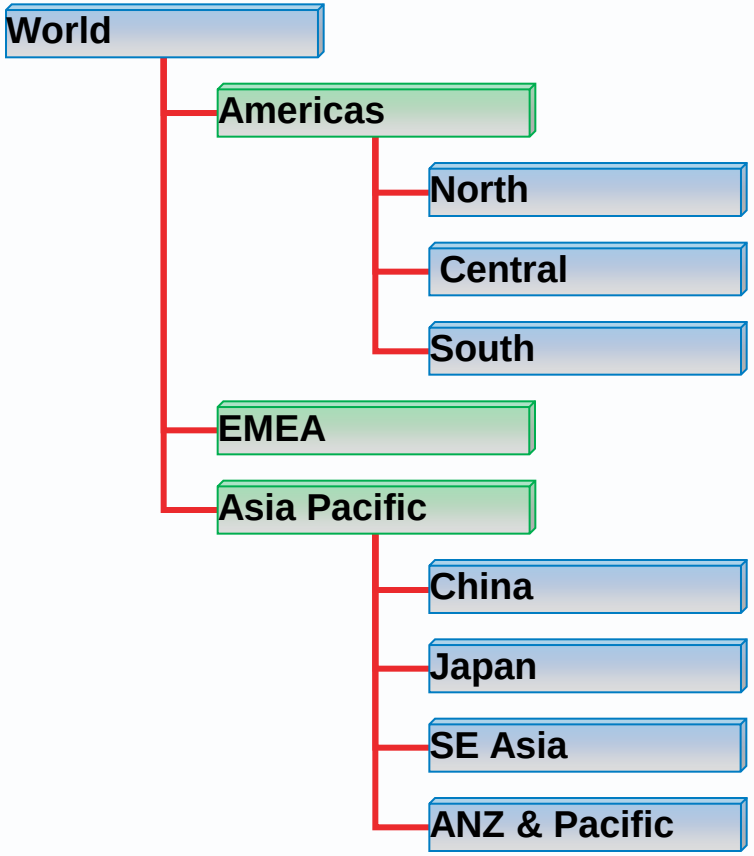


One person's how is another's what



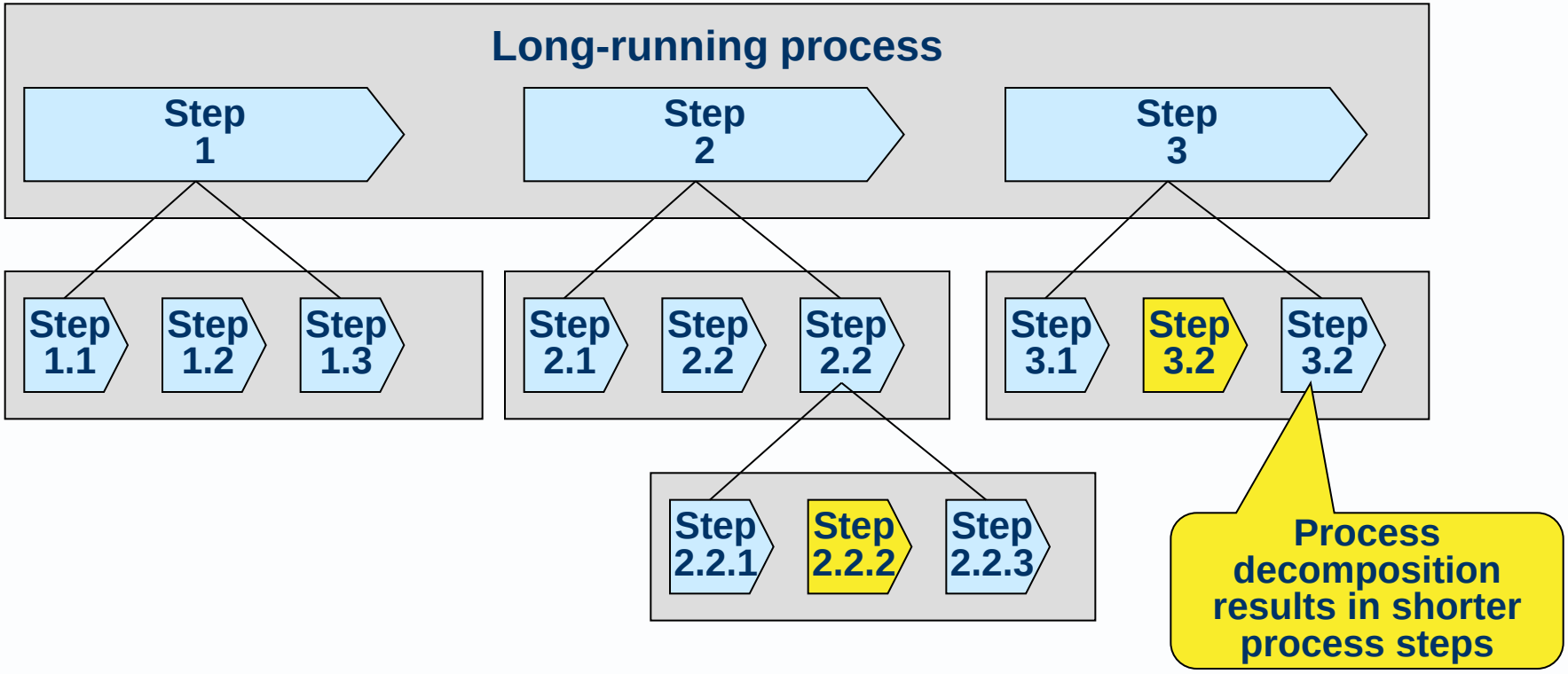
3. Geographical organisation

- Described hierarchically to help locate things



4. Decomposition of system behaviour into process steps

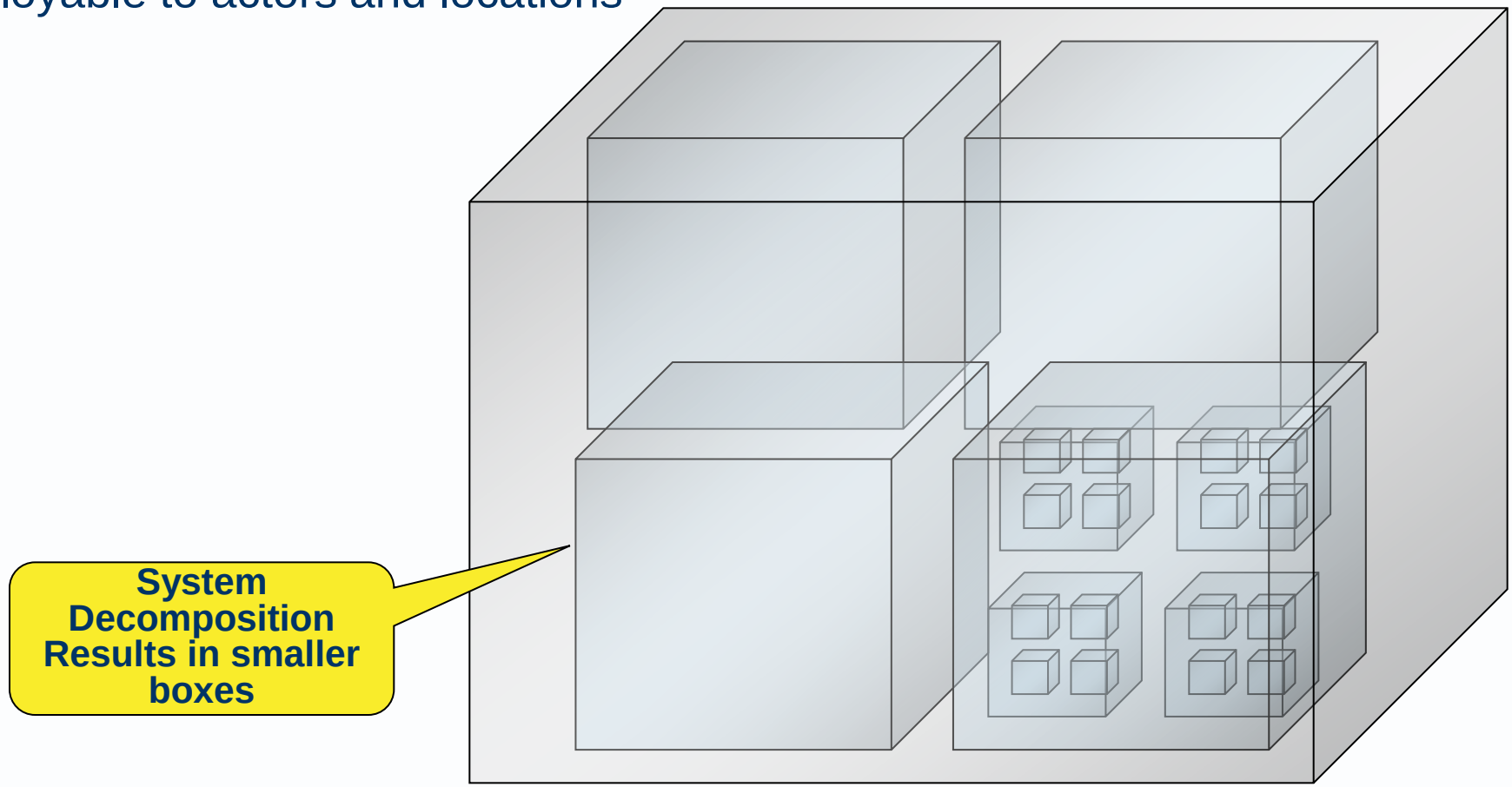
- To analyse requirements, and to define automatable processes.



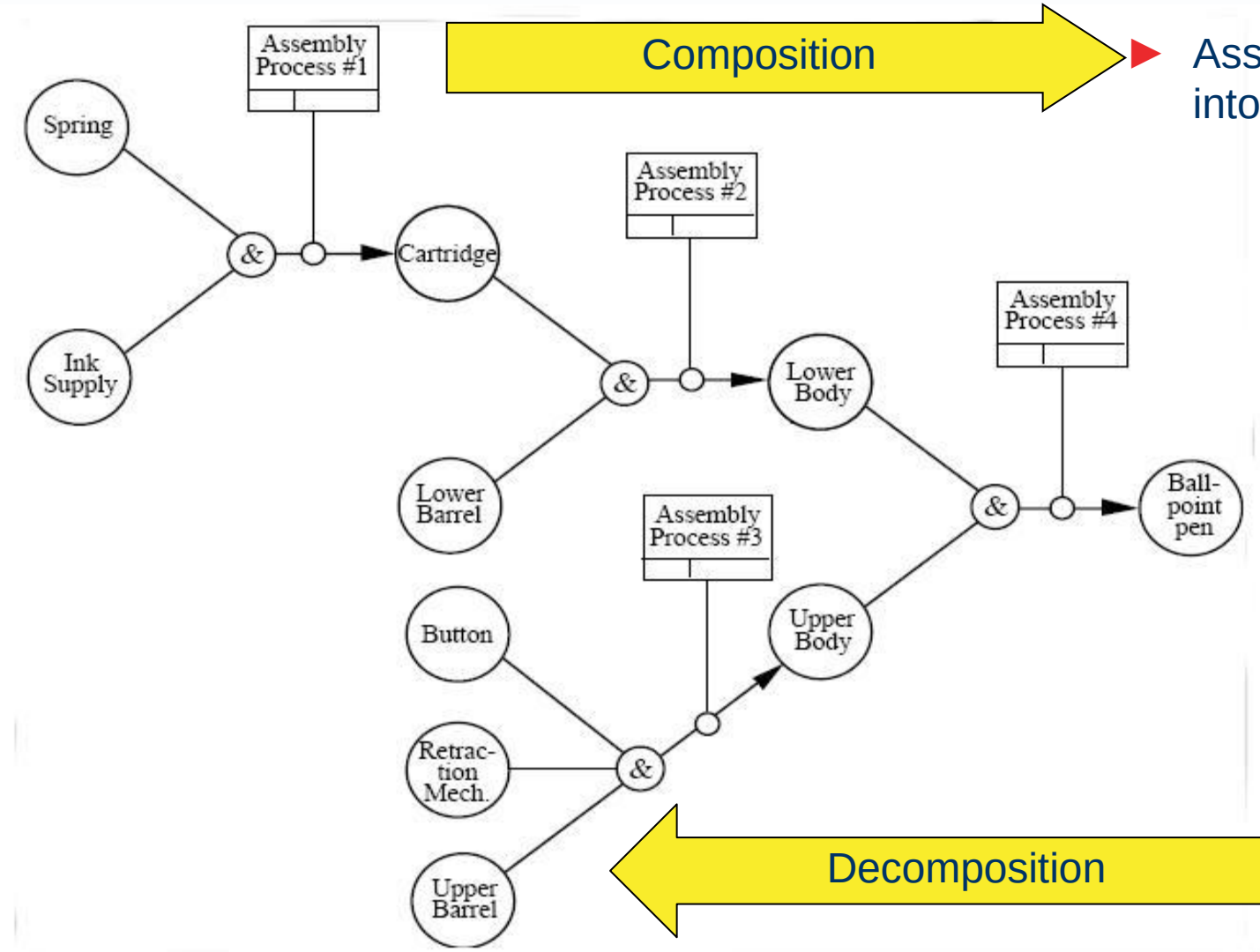
- Look out for reusable sub-processes

5. Decomposition of system structures into components

- to make systems buildable
- deployable to actors and locations



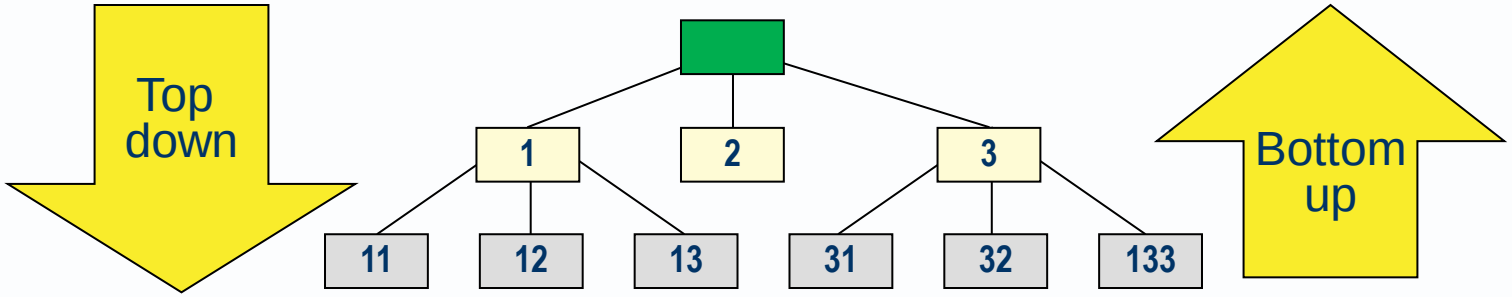
Composition – illustrated using IDEF 5



- Assembling parts into a whole
- Dividing a whole into parts

Cycling top down and bottom up design

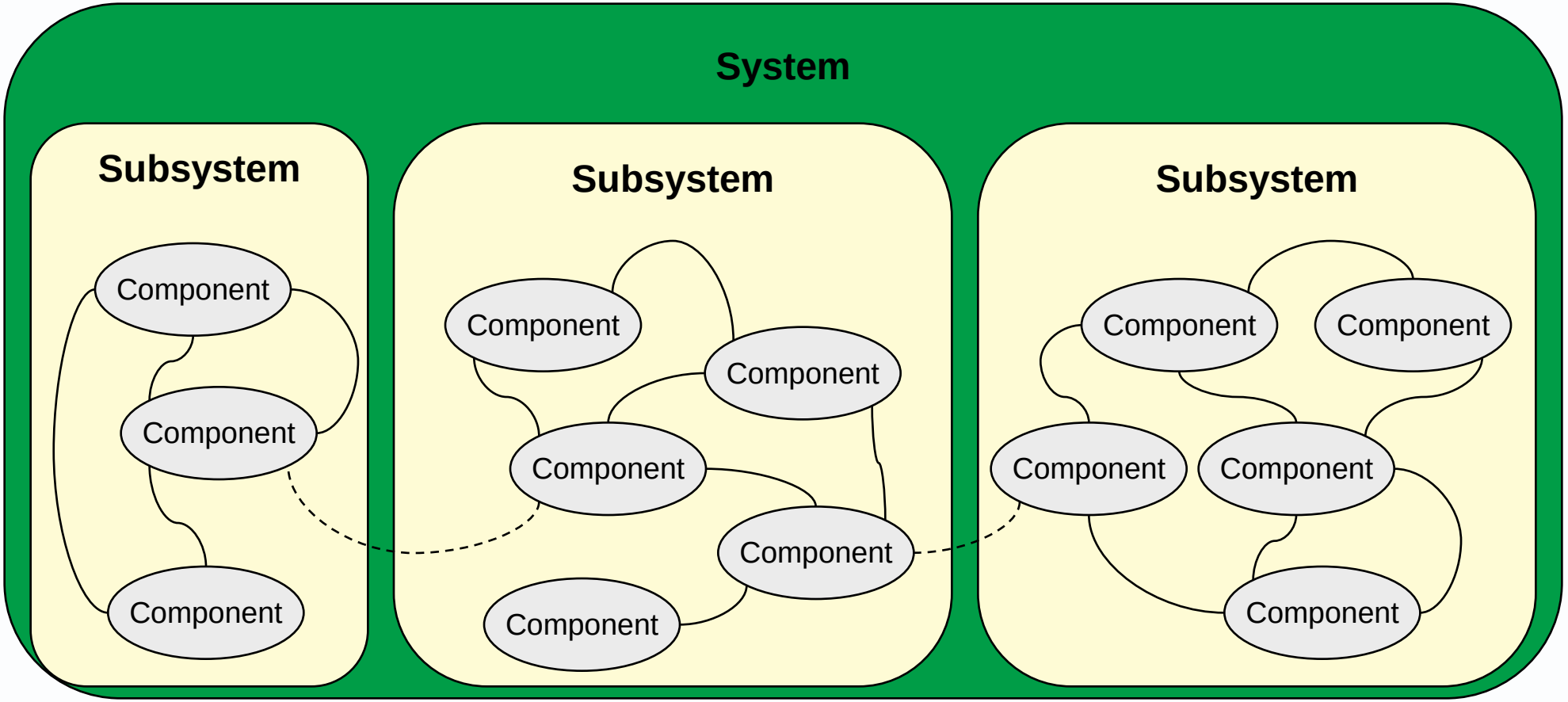
- A structure defined from the top down is rarely optimal
- So alternate top down and bottom up design



- This helps to ensure that a higher level model is well formed:
  - a manageable and accurate abstraction of lower levels
  - contains elements at a consistent level of granularity
  - shows what is more important to the viewer
  - hides what is less important to the viewer

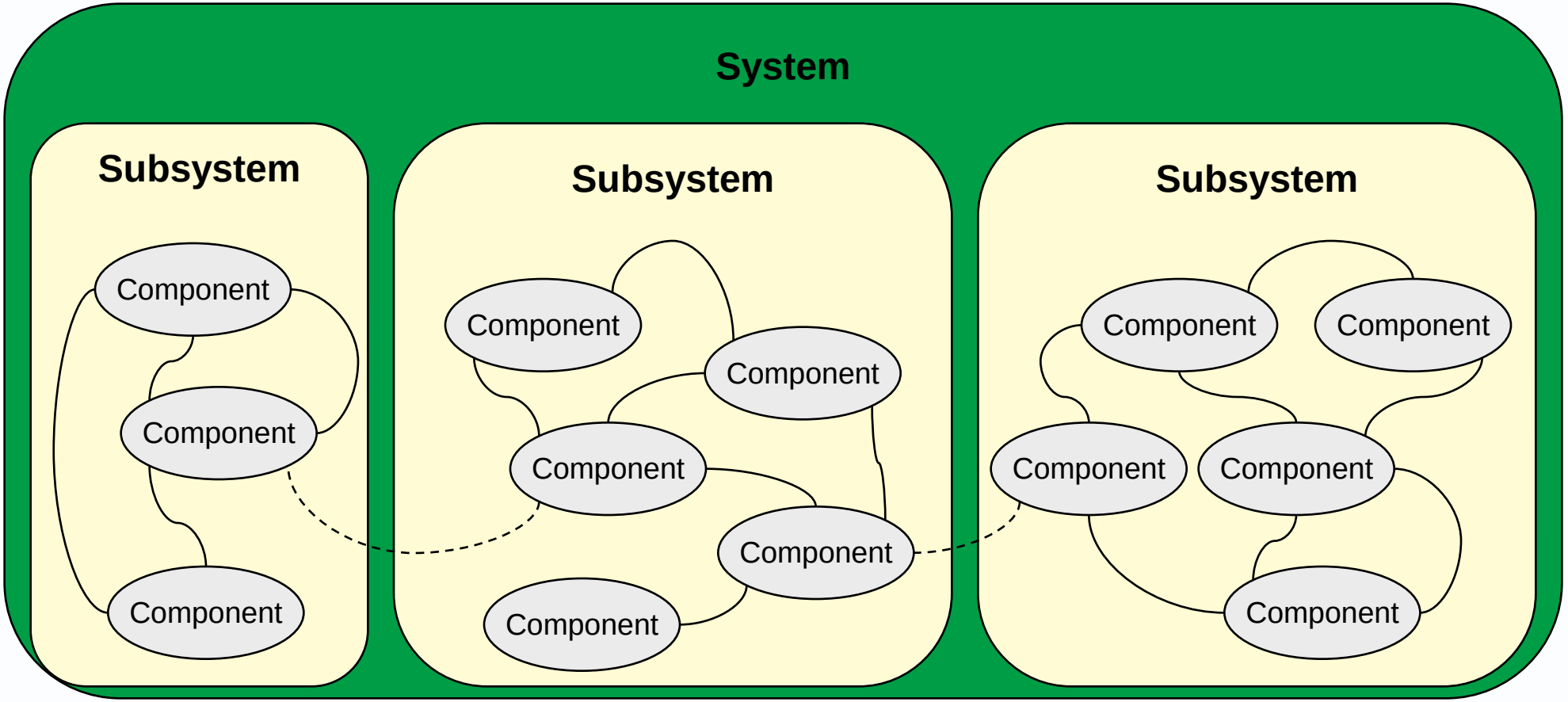
To hide details you can..

- Cluster related components into a subsystem.



To make subsystems independently manageable…

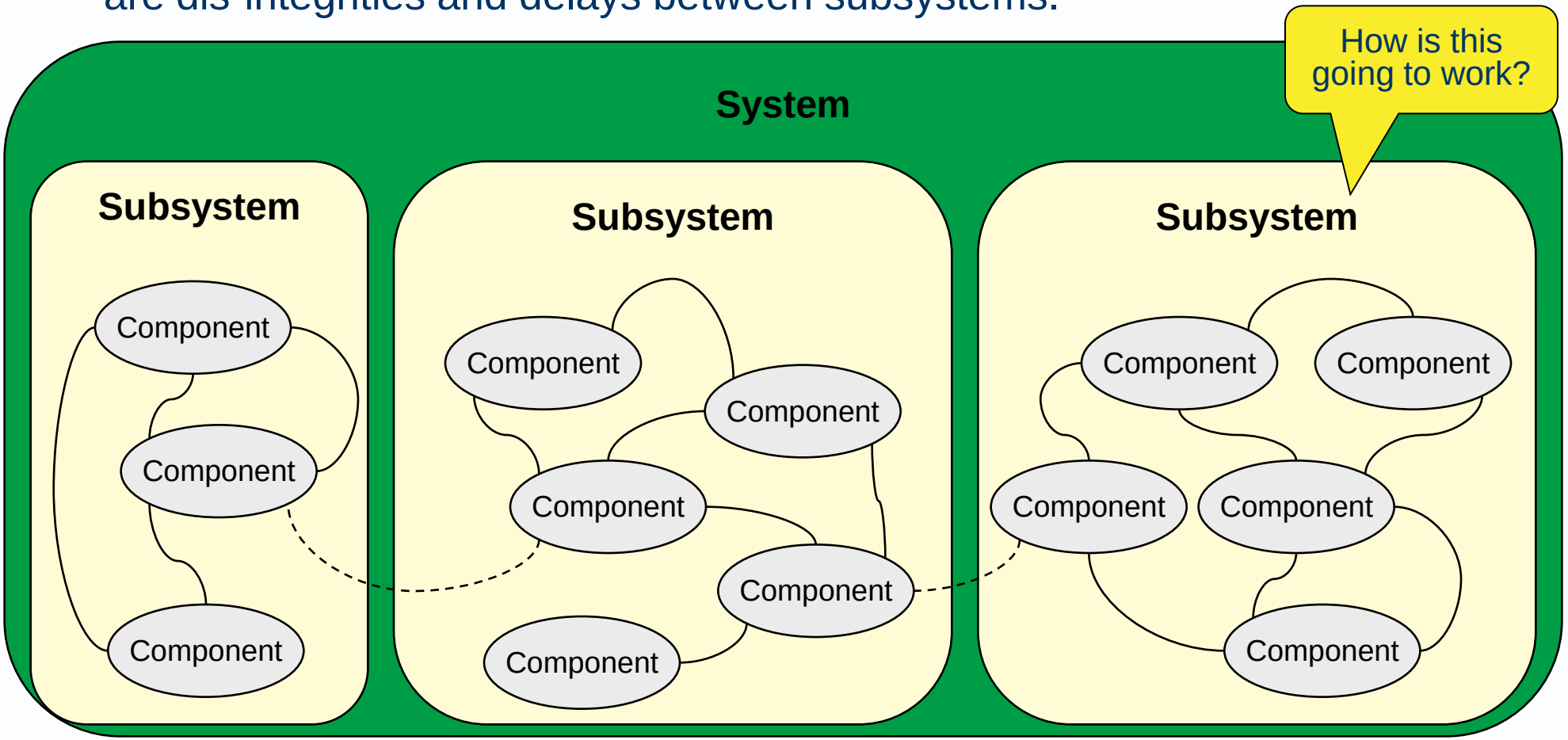
- gather ***cohesive*** components in one subsystem, and
- separate ***loosely-coupled*** components in different subsystems.



Separating subsystems makes them more independent and flexible

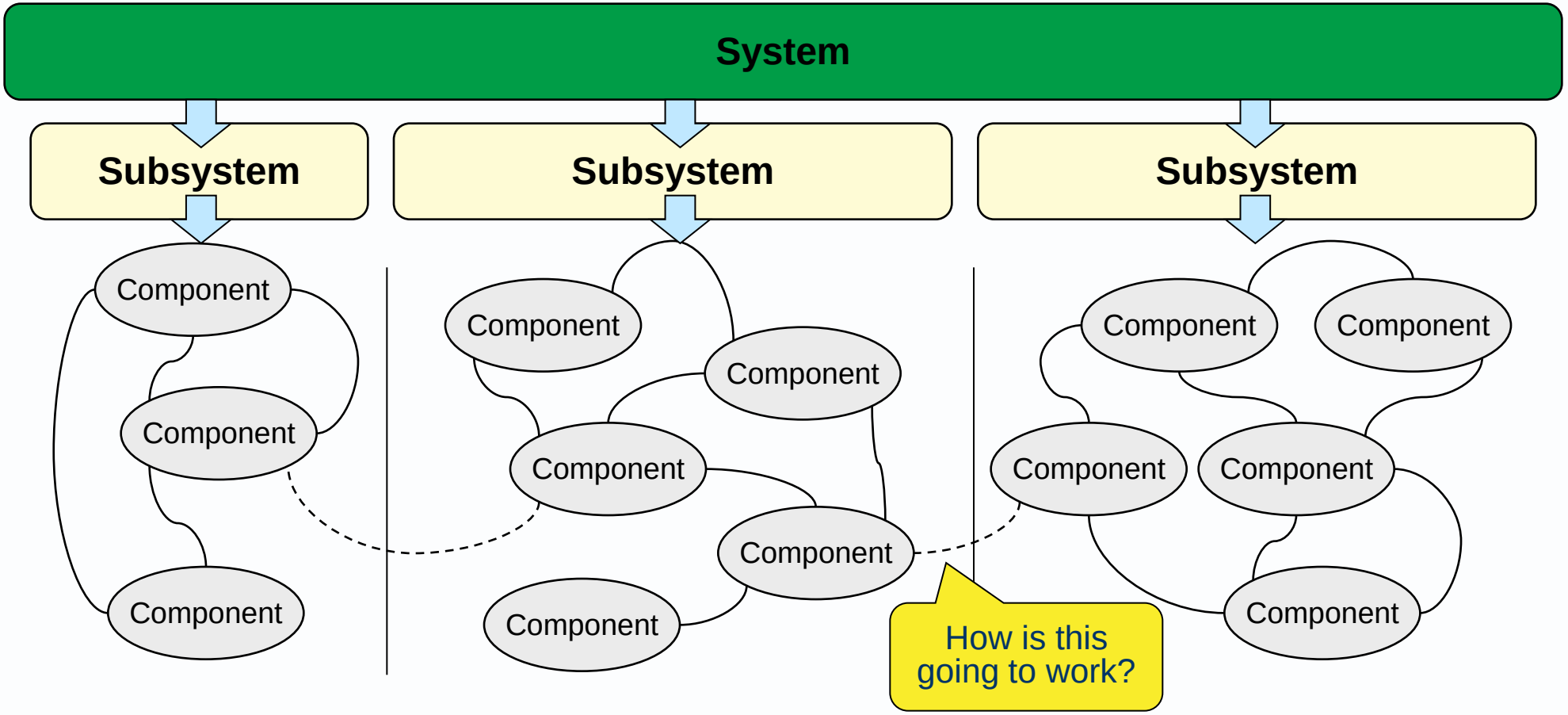
Avancier

- But loosely-coupled subsystems can be inconsistent, since there are dis-integrities and delays between subsystems.



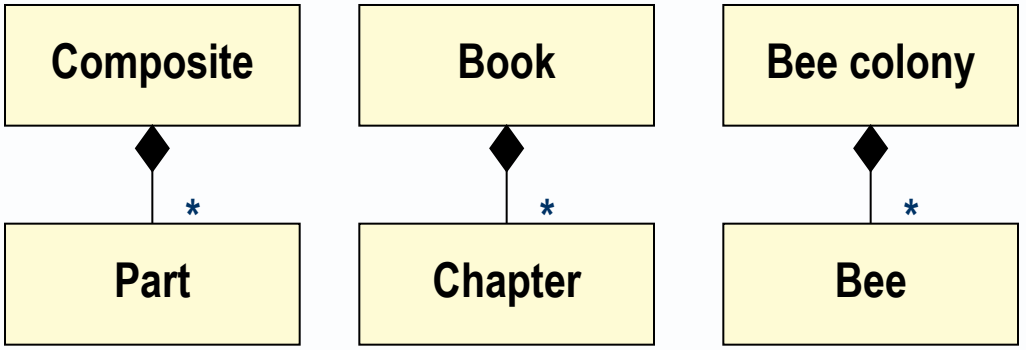
How is a hierarchical structure implemented in reality?

- Higher level components may act as *façades or managers* that delegate work to lower level or inner components.



Composition: definition

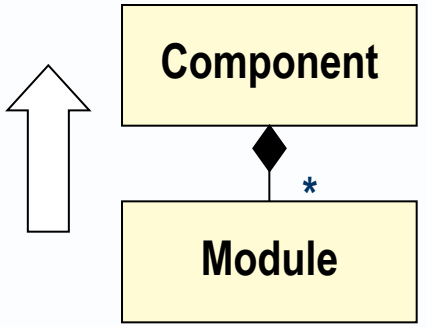
- **Composition:** Assembly of parts into a whole and/or hiding internal component behind an interface
- Makes a thing or description more manageable
- Encapsulates content in a shell, or organises units under a manager.



- **Decomposition:** Division of a whole into parts and/or identification of components behind an interface
- Makes a description more detailed and useful to a builder.

In short

- Composition creates a larger, more coarse-grained, thing
- You can hide what is inside (as far as possible)
- And describe the composite more abstractly



Avancier Methods (AM)

Concepts

A few remarks on abstraction

Enterprise architects use abstraction

- Since business systems contain millions of elementary parts, EAs tend to catalogue coarse-grained composites

Composition	Data	Applications	Infrastructure
Coarse-grained	**Data store (CRM)**	**Application (CRM)**	**Network Domain**
Mid-grained	Aggregate entity (Opportunity)	Component/Package	Virtual server
Fine-grained	Data entity (Quote)	Module/Class	Execution environment
Elementary part	Data item (Price)	Operation/Method	Executable
Decomposition	**Data element**	**Software element**	**Deployable element**

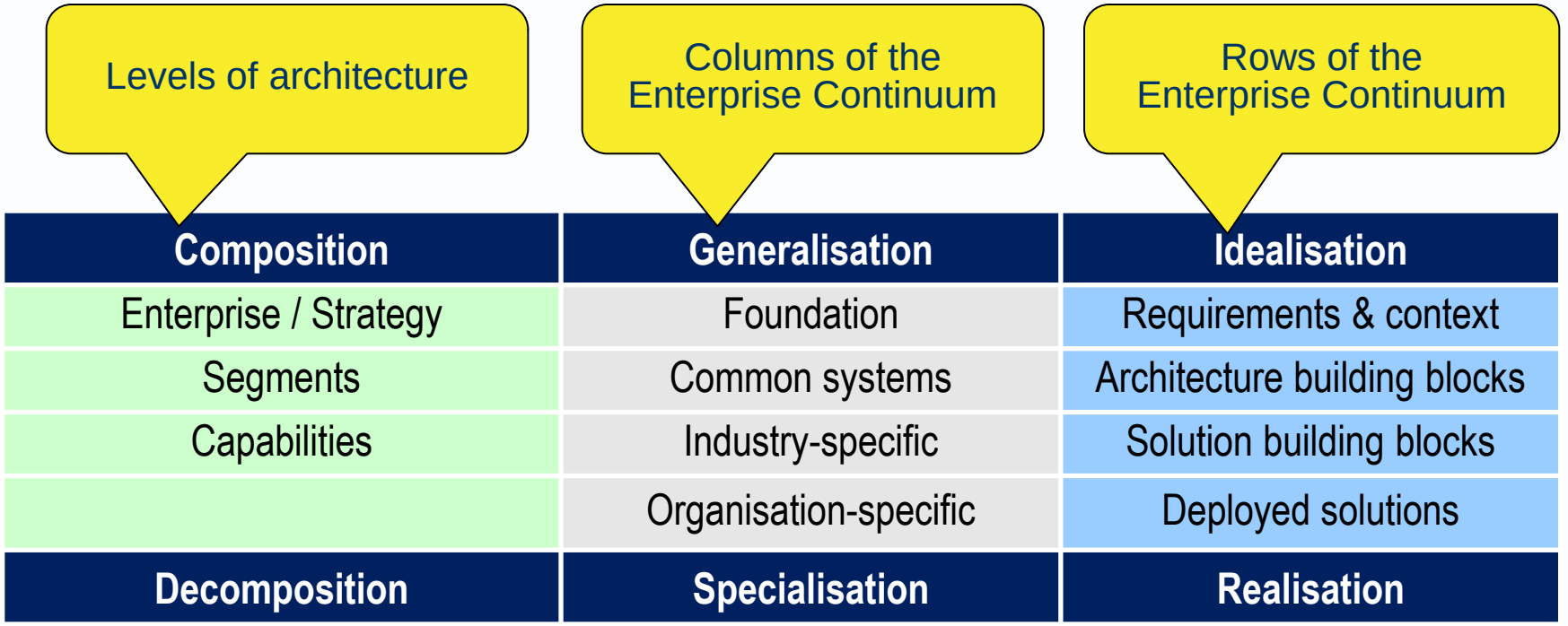
Enterprise architects use several kinds of abstraction

- EAs use coarse-grained and/or generic and/or conceptual views

To manage large systems	To reuse structures and processes	To speak with directors and managers
Composition	**Generalisation**	**Idealisation**
Coarse-grained composite	**Universal**	**Conceptual Model**
Mid-grained composite	**Fairly generic**	**Logical Model**
Fine-grained composite	Fairly specific	Physical Model
Elementary part	Uniquely configured	Physical Material
Decomposition	**Specialisation**	**Realisation**

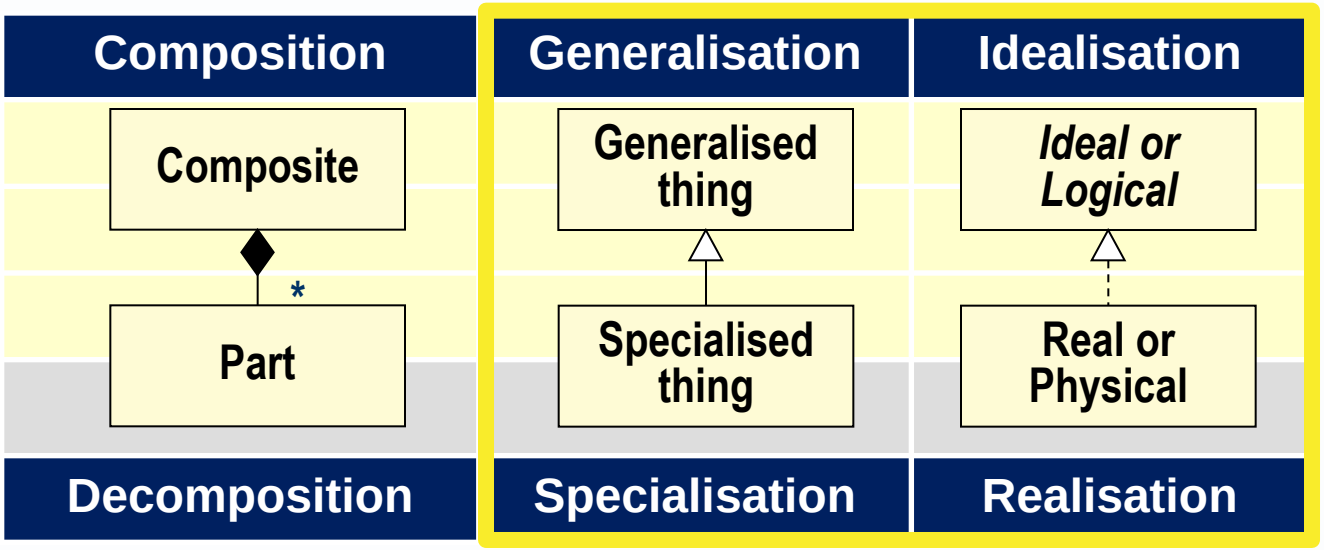
E.g. TOGAF uses all three kinds of abstraction

- To classify and divide up architecture descriptions.



We'll discuss other kinds of abstraction later

- More to come on Generalisation and Idealisation



Note, the abstract must connect to the concrete

- EA is not purely philosophical
- EA who stay at an abstract level risk missing cultural and physical design issues that are show stoppers
- EA need to understand and govern the implications of rolling out strategic decisions such as
  - Integrate Order Management and CRM systems
  - Deploy one identity management system world-wide
  - Standardise customer name and address data types
  - Migrate hand-made spreadsheets into a central ERP system