Avancier Methods (AM)
DOCUMENT

Generalities about the diagrams used in an Avancier's Model Language (AML)

It is illegal to copy, share or show this document
(or other document published at http://avancier.co.uk)
without the written permission of the copyright holder

Copyright Avancier Limited

Architecture documentation framework - contents

Architectural entities	POLDAT etc.
Architecture artefacts	**Catalogues** that list architectural entities with attributes
	Matrices that relate architecture entities
	Diagrams that describe and/or relate architectural entities
Architecture models and languages	e.g. ArchiMate, UML, AML
Management documents	Deliverables, work in progress or signed off. Often contain architecture artefacts
A meta model for an architecture repository	Defines architectural entities, their attributes and inter-relationships
Pre-defined classifications and reference models	Generalised taxonomies and common design patterns

Copyright Avancier Limited

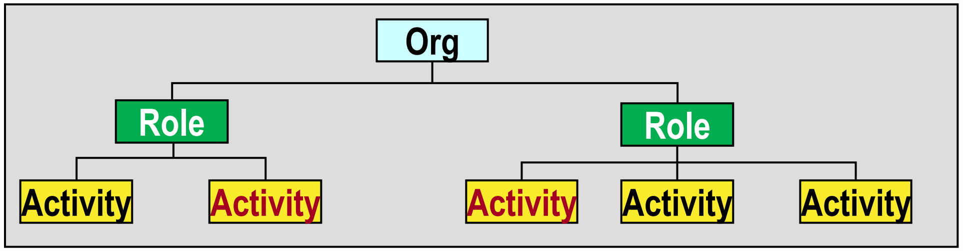
Ideas to consider

Ideas to consider
Defining your own symbols
Agree the meta model of your architecture description

Copyright Avancier Limited

Hierarchical catalogues to matrices to network diagrams

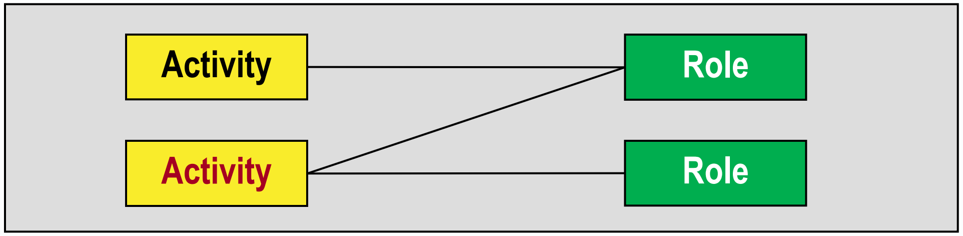
- A hierarchy describes one to many relationships well, but many to many relationships poorly



- Where a hierarchy as many duplicated elements, a matrix is better

	Role	Role
Activity	X	
Activity	X	X

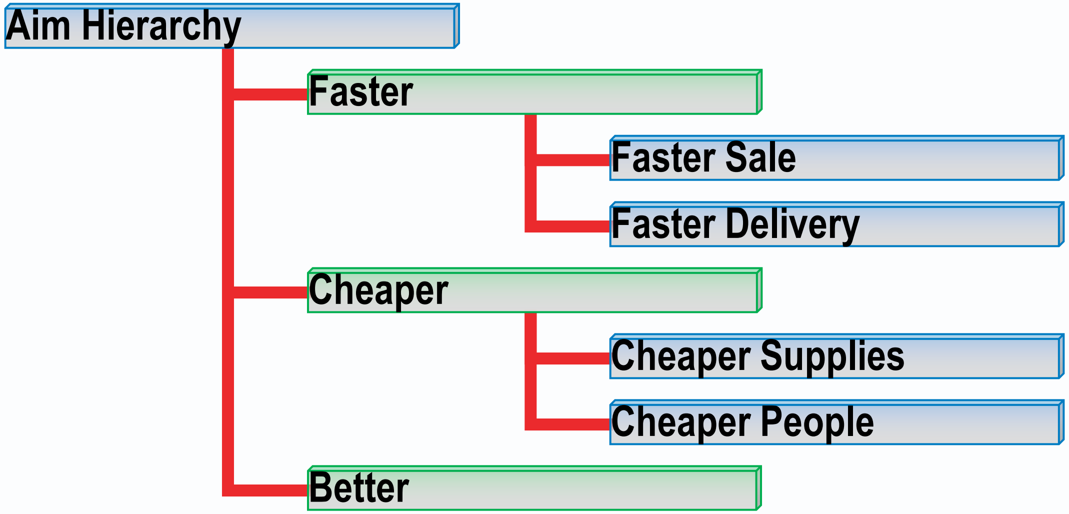
- Where a matrix is large and empty, a network diagram is better



Copyright Avancier Limited

You can turn catalogues into trees and tables

- Trees are good for showing one-to-many cascades



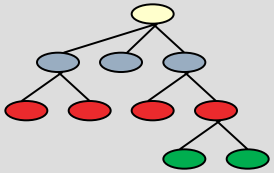
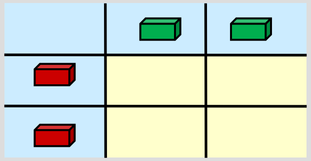
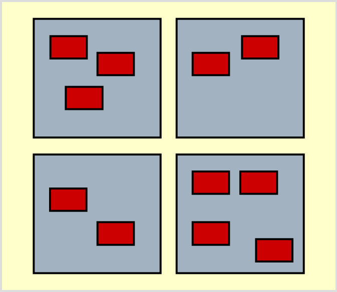
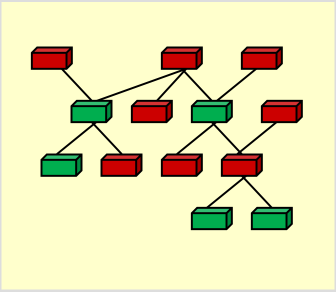
- Tables are better for showing many-to-many relationships

Application Role	CRM	ERP	Billing	Data Warehouse
PR officer	Uses services of			Uses services of
Salesman	Uses services of			Uses services of
1st line support	Uses services of		Uses services of	Uses services of
Fulfilment agent	Uses services of	Uses services of		

Copyright Avancier Limited

You can turn trees and tables into colored-box diagrams

- Hierarchies - nested colored boxes
- Network structures - colored box-line diagrams

	Hierarchies	**Networks**
Plain structures	Tree 	Table 
Colored box diagrams	Nested boxes 	Node-arc diagram 

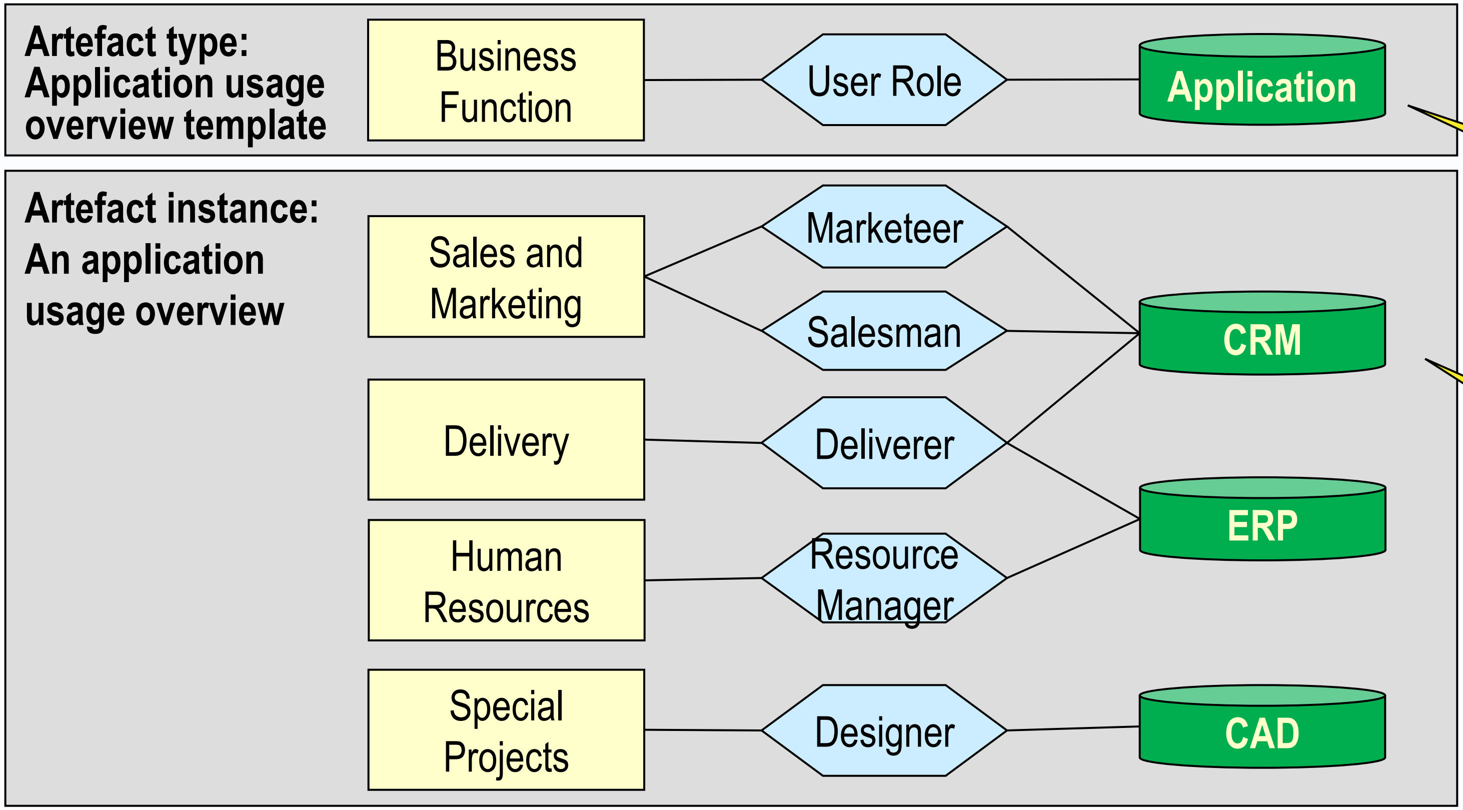
Architect often devise their own node-arc diagrams for

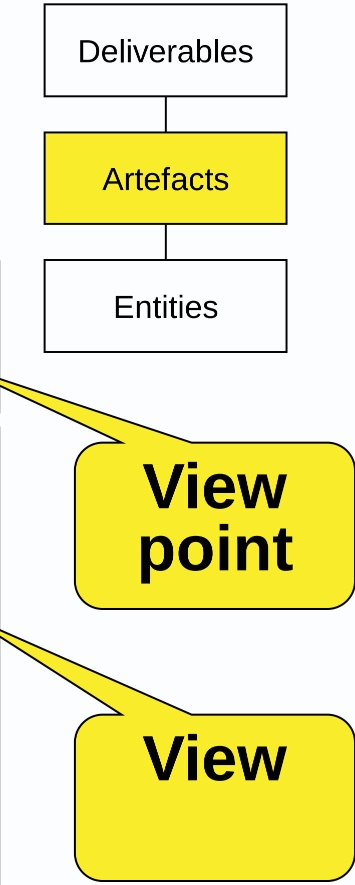
- structural models of components
- behavioural models of processes.

Copyright Avancier Limited

You can show N-way relationships as diagrams

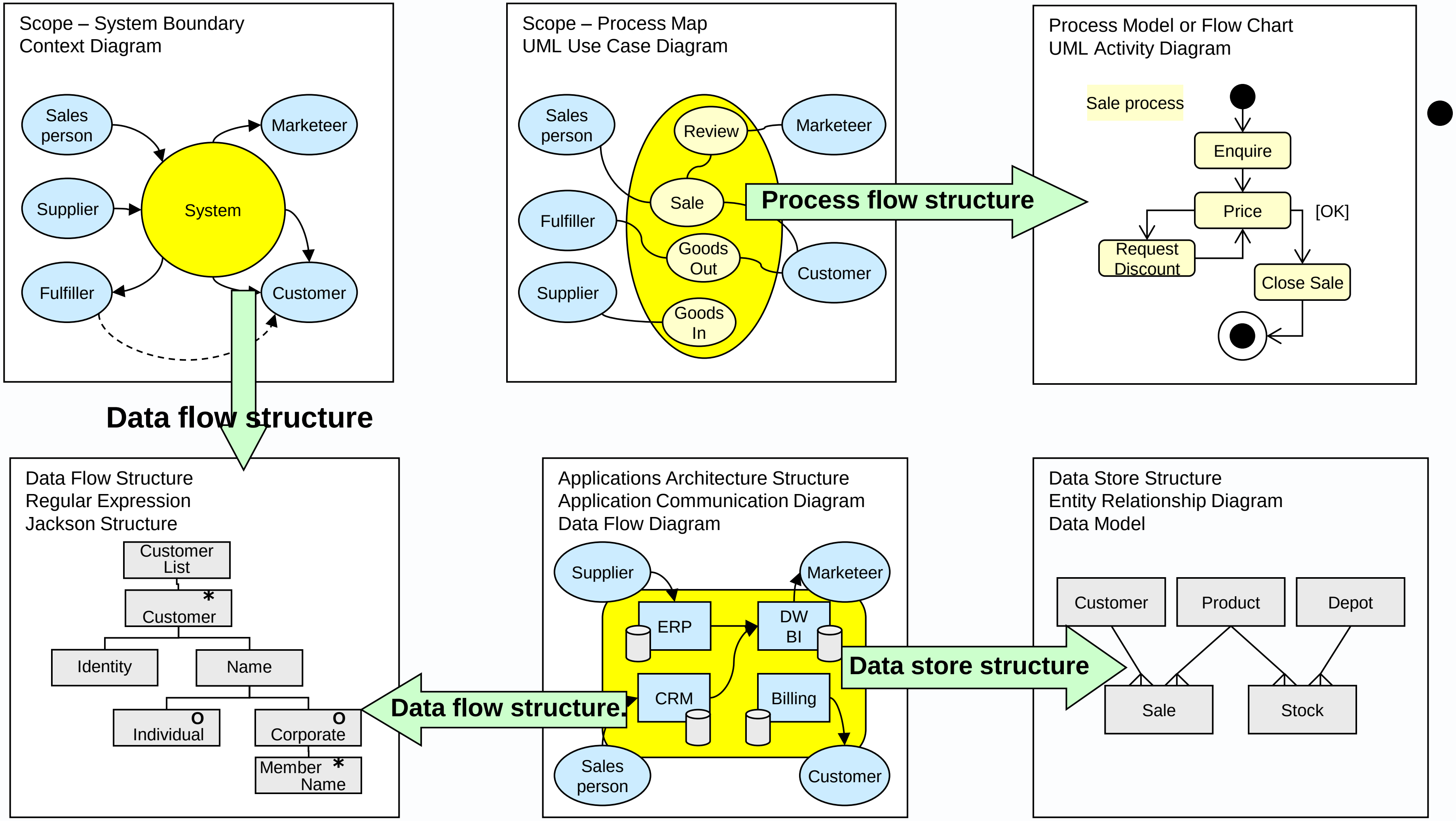
- An enterprise architect can go a long way with
  - Hierarchical structures of Entities
  - Mappings between Entities shown in Matrices
- But a diagram is often more economical and easier to validate.





Copyright Avancier Limited

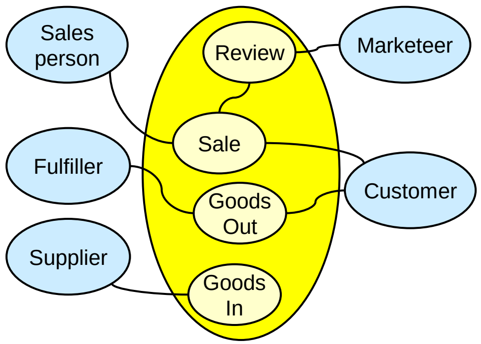
Diagrams form inter-related views of an architecture



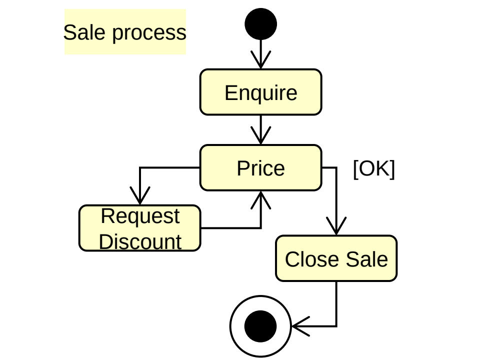
Copyright Avancier Limited

A range of diagram types

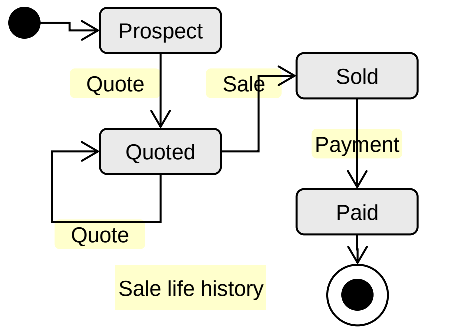
Scope – Process Map
UML Use Case Diagram



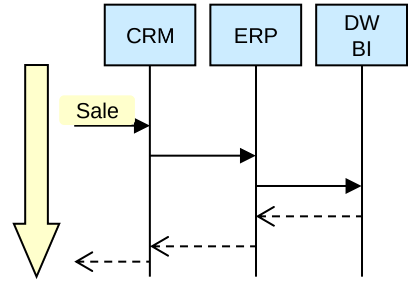
Process Model or Flow Chart
UML Activity Diagram



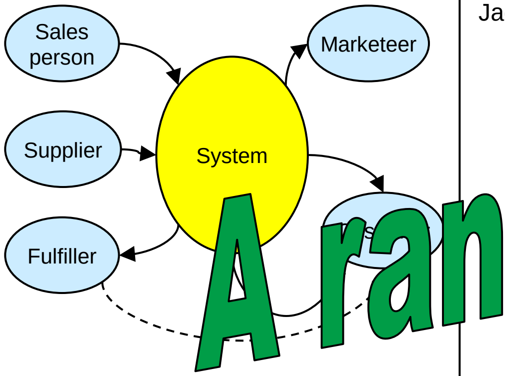
Data Entity Life Cycle
State Transition Diagram
UML State Chart



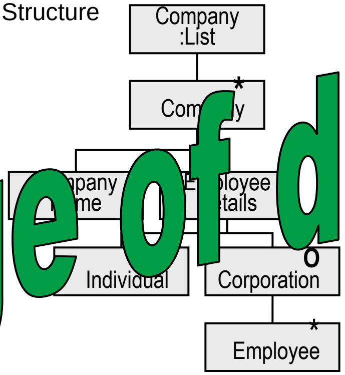
Applications Architecture Behaviour
Process System Realisation
UML Interaction/Sequence Diagram



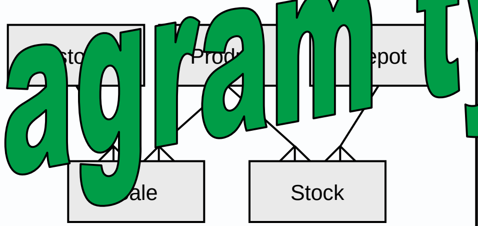
Scope – System Boundary
Context Diagram



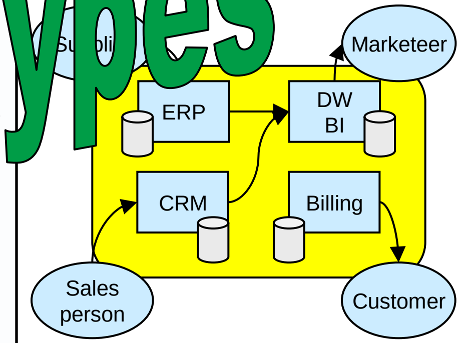
Data Flow Structure
Regular Expression
Jackson Structure



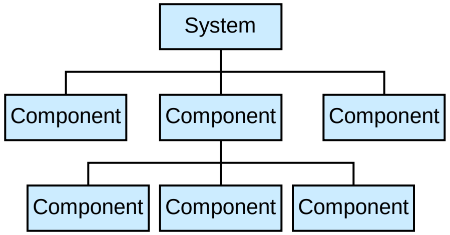
Data Store Structure
Data Model
Entity Relationship Diagram



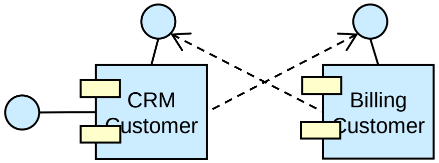
Applications Architecture Structure
Application Communication Diagram
Data Flow Diagram



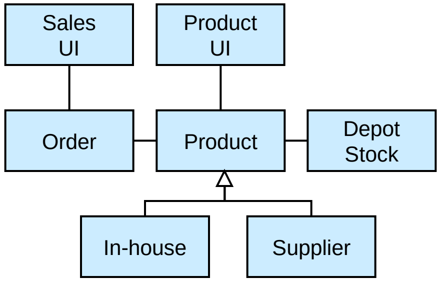
Hierarchical Structure
System or Process Decomposition



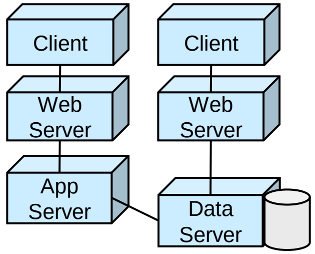
Components, Interfaces and Dependencies
UML Component Diagram



Software Architecture Structure
UML Class Diagram



Infrastructure Architecture Structure
Hardware Configuration Diagram
UML Deployment Diagram



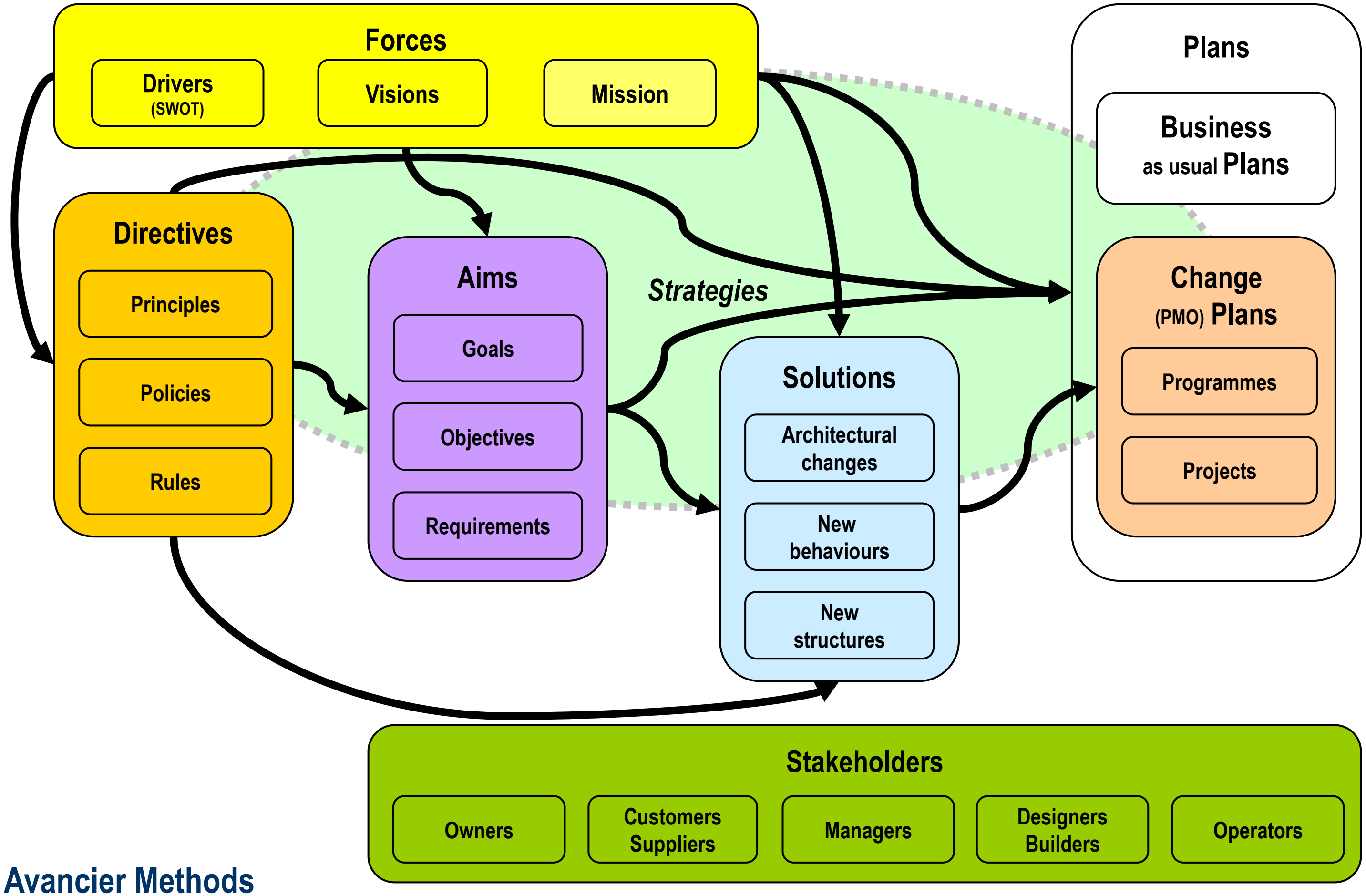
A dozen diagrams

Copyright Avancier Limited

Used to define system structure and behaviour

Record the motivations for architecting

- Anything in this context and among these precursors may be annotated on any architectural model

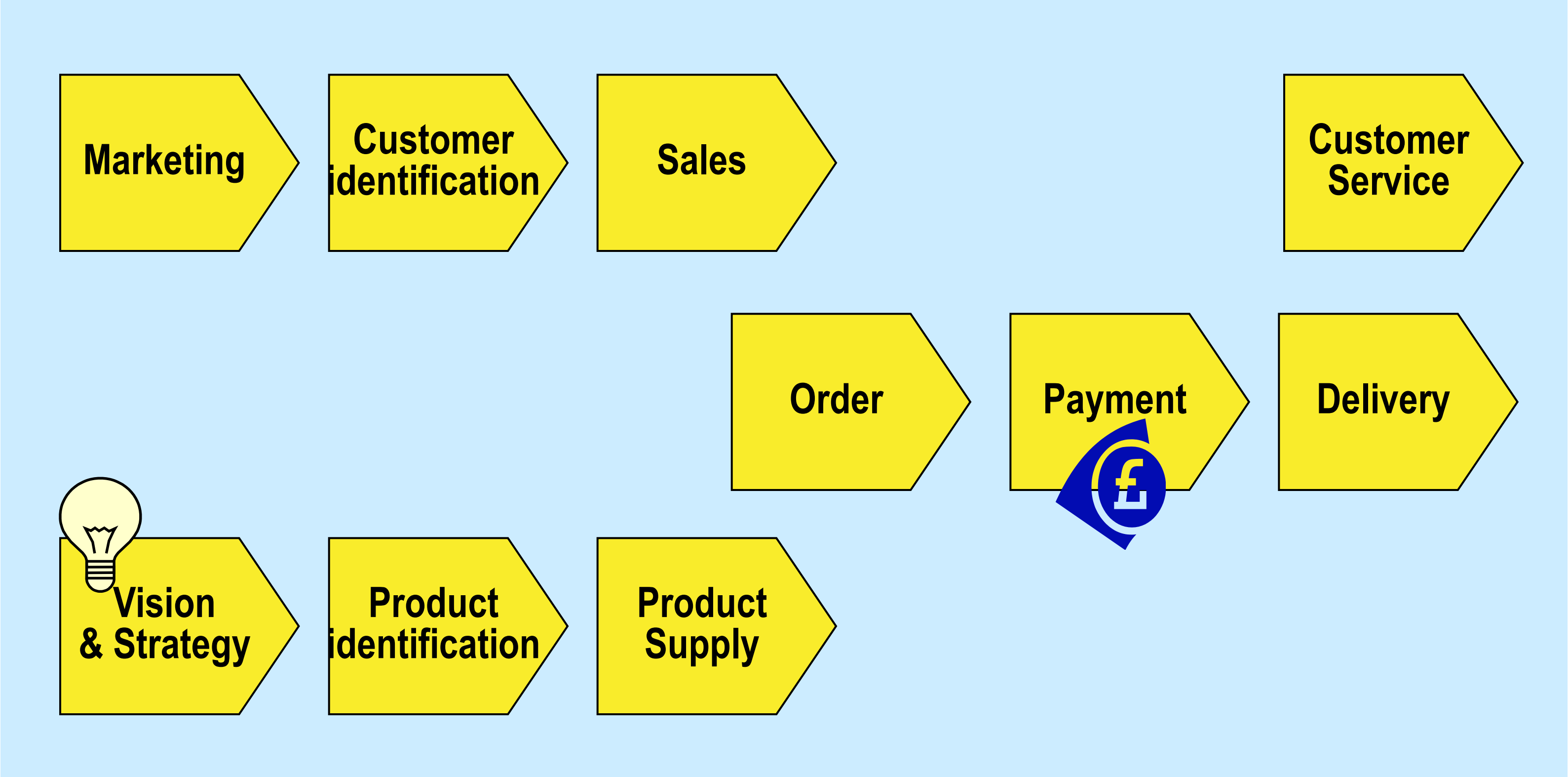


Avancier Methods

Copyright Avancier Limited

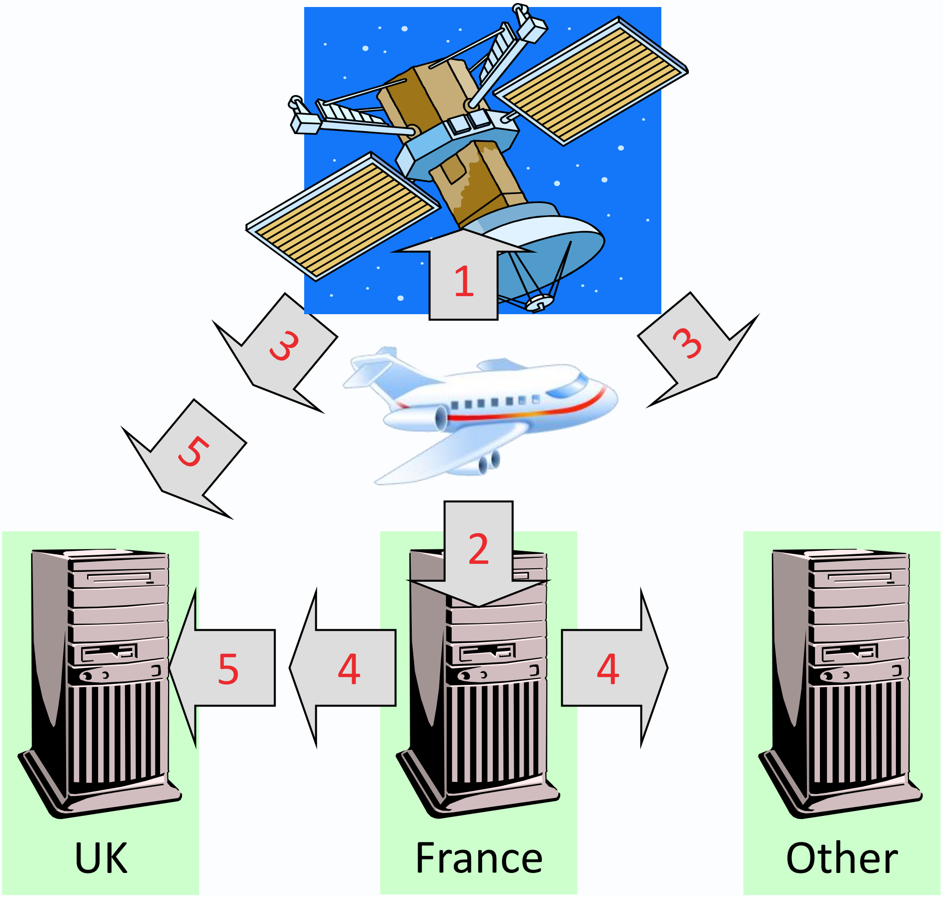
Draw cartoons where relationships are vaguely-defined

- E.g. to show business functions in a “value chain/stream”, don’t pretend you are drawing a flow chart with step-to-step transitions. Draw a cartoon instead.



Use rich pictures to express a solution vision

- To express a solution vision



Copyright Avancier Limited

Support structural models with behavioural models

Structural models

- Component diagram
  - components of a system and
  - their interrelationships
    - be they service flows, data flows or dependencies.
- Process map:
  - processes that actors of a system are involved in, and
  - dependencies between processes

Behavioural models

- Process flow chart
  - step-by-step flow of work - shows the overall flow of control.
- Sequence diagram
  - a sequence of message-based interactions between components.

Copyright Avancier Limited

Defining your own symbols

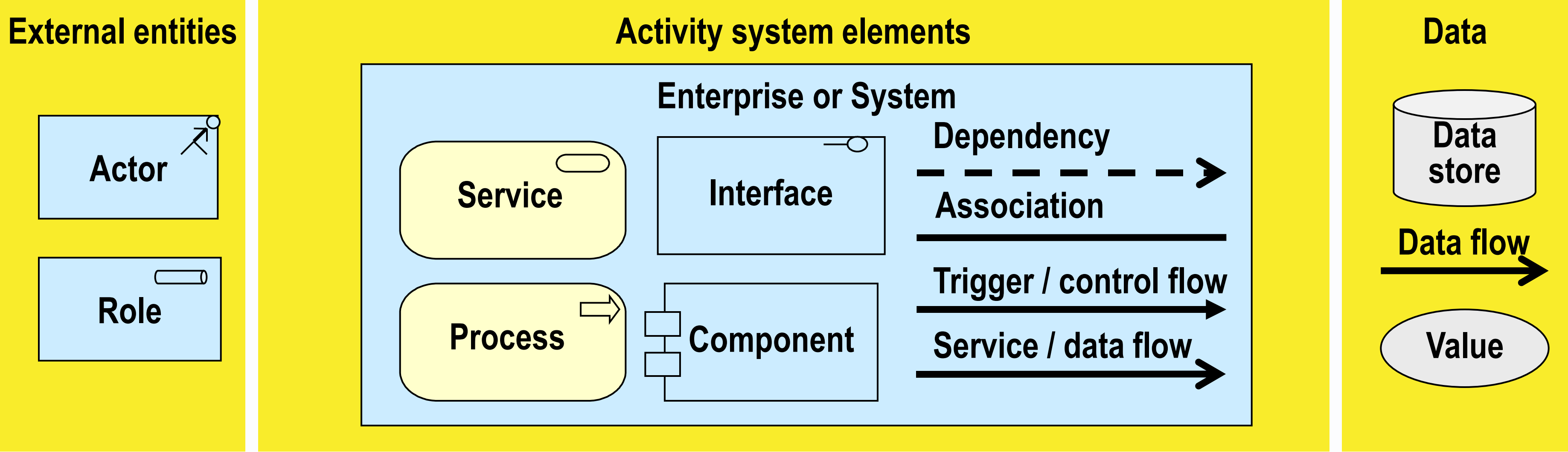
Ideas to consider

Defining your own symbols

Agree the meta model of your architecture description

Copyright Avancier Limited

Define box symbols for your architectural entities



External entities

Actor

Role

Activity system elements

Enterprise or System

Service

Interface

Process

Component

Dependency

Association

Trigger / control flow

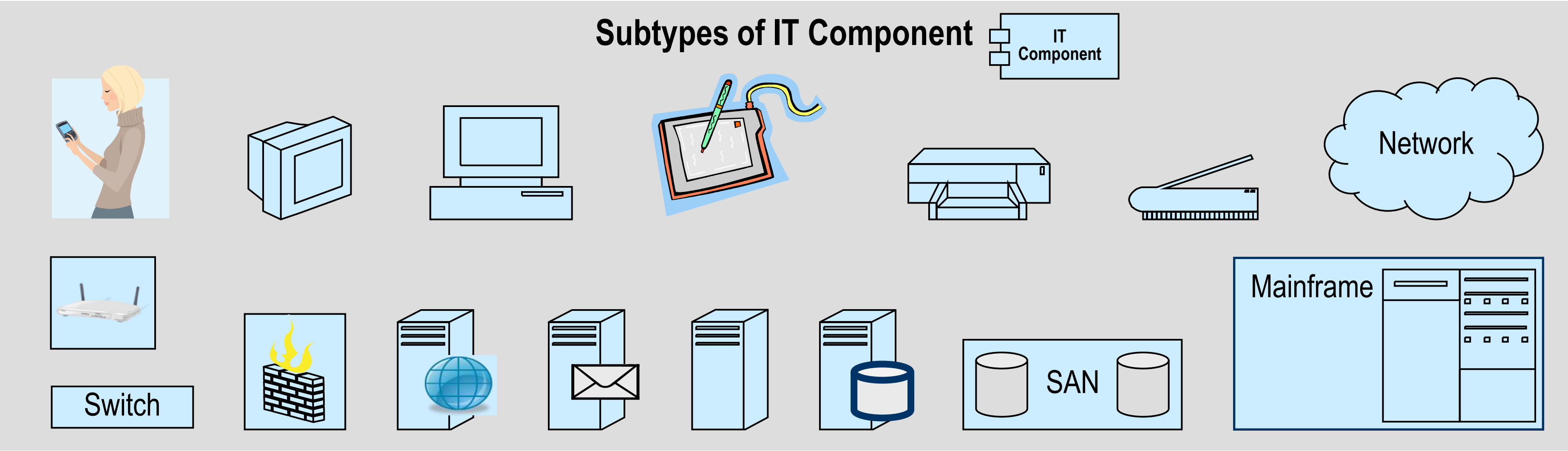
Service / data flow

Data

Data store

Data flow

Value



Subtypes of IT Component

IT Component

Network

Switch

SAN

Mainframe

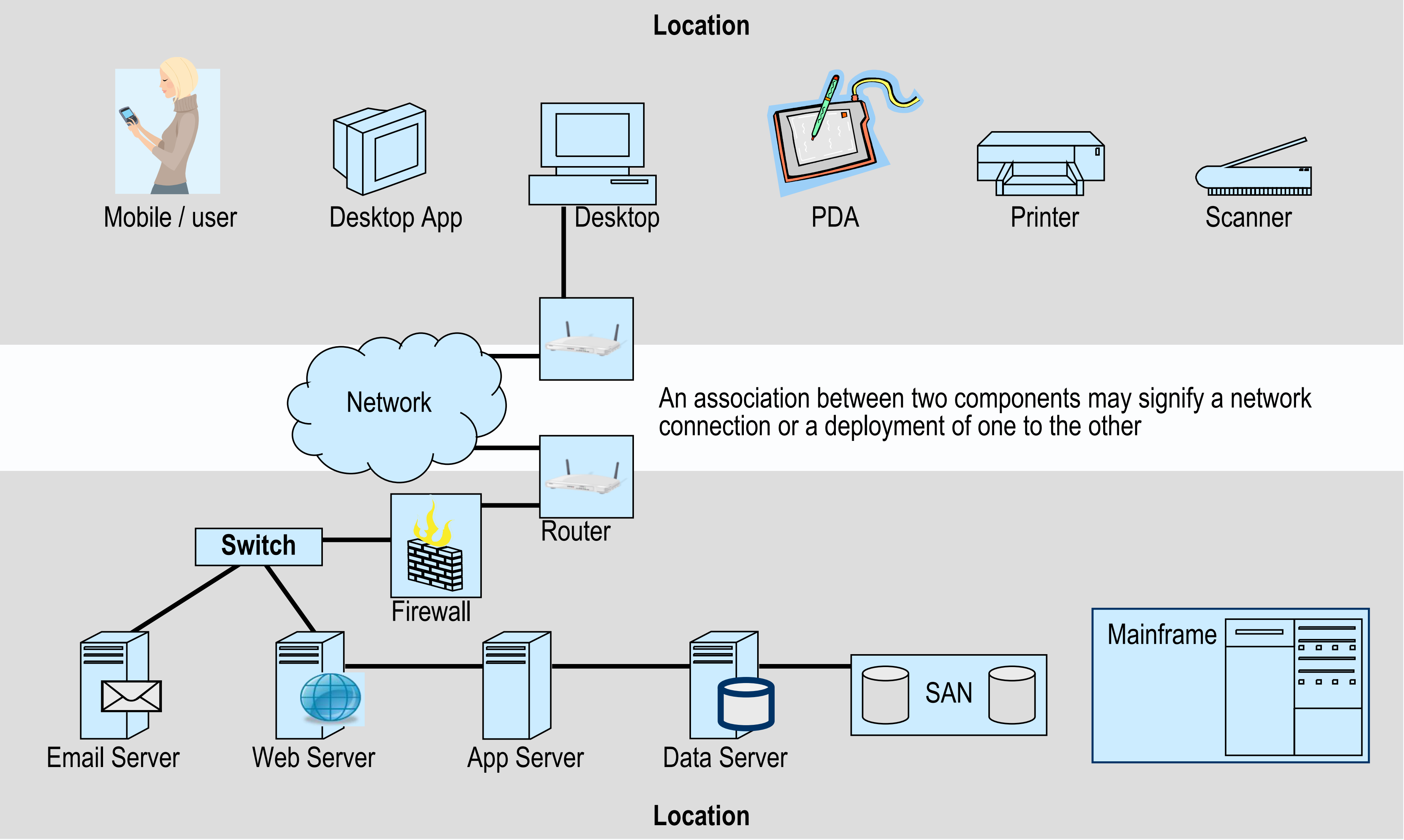
Copyright Avancier Limited

Define line symbols for relationships between architectural entities

Meaning of relationship	Line symbol
Part-Composition	──◆
Specialisation-Generalisation	──▷
Realisation-Idealisation	- - ▷
Association	───
Trigger or control flow	──▸
Service or data flow	──>
Dependency	- - >

Copyright Avancier Limited

Remember the architecture is in the relationships

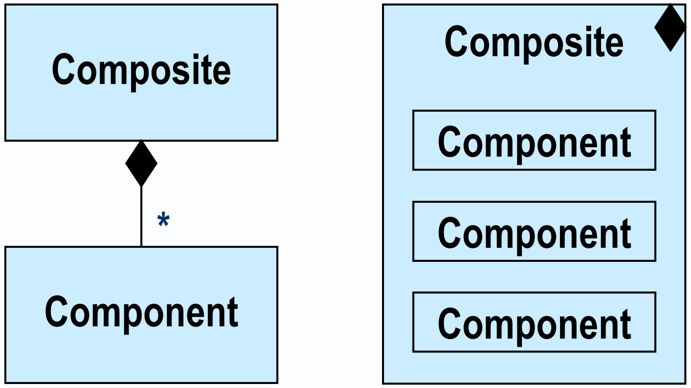


An association between two components may signify a network connection or a deployment of one to the other

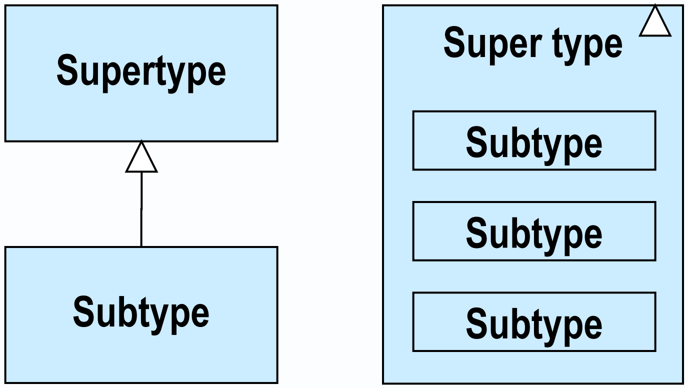
Copyright Avancier Limited

Define how you will model abstraction

- Abstraction by composition



- Abstraction by generalisation



Copyright Avancier Limited

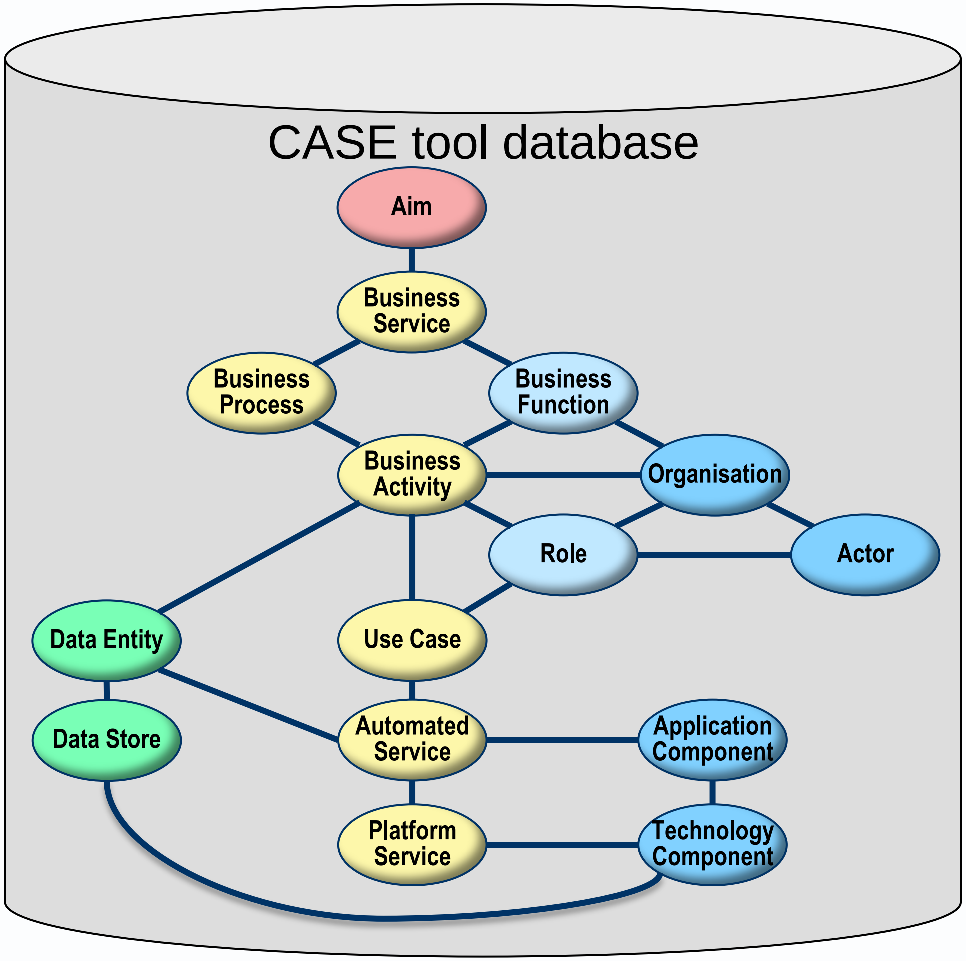
Agree the meta model of your architecture description

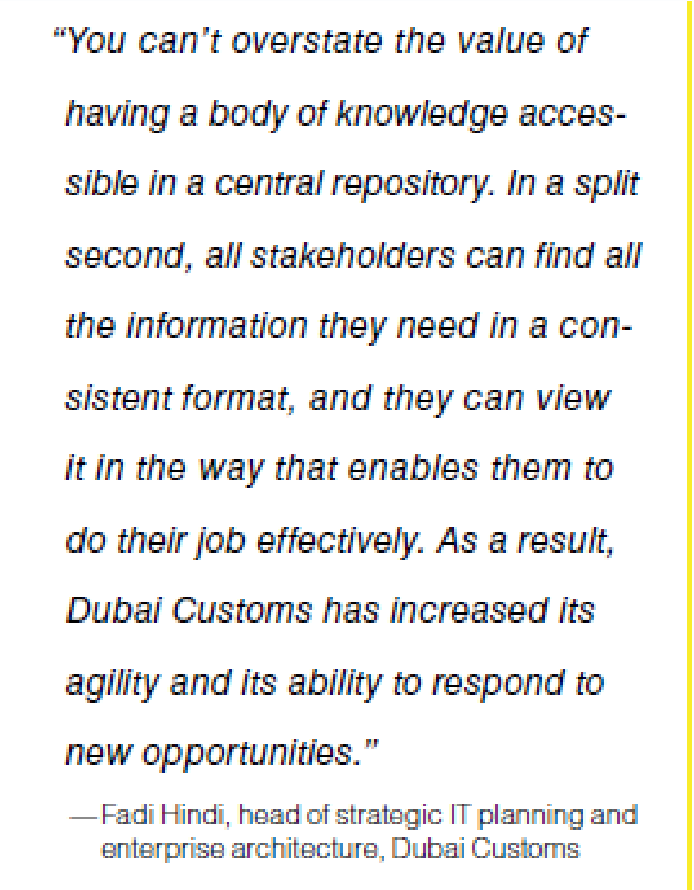
- Ideas to consider
- Defining your own symbols
- Agree the meta model of your architecture description

Copyright Avancier Limited

It helps to record your architecture in a repository

- Clinger Cohen Act 1996 says a Fed. Gov. Agency must “maintain an IT architecture repository”



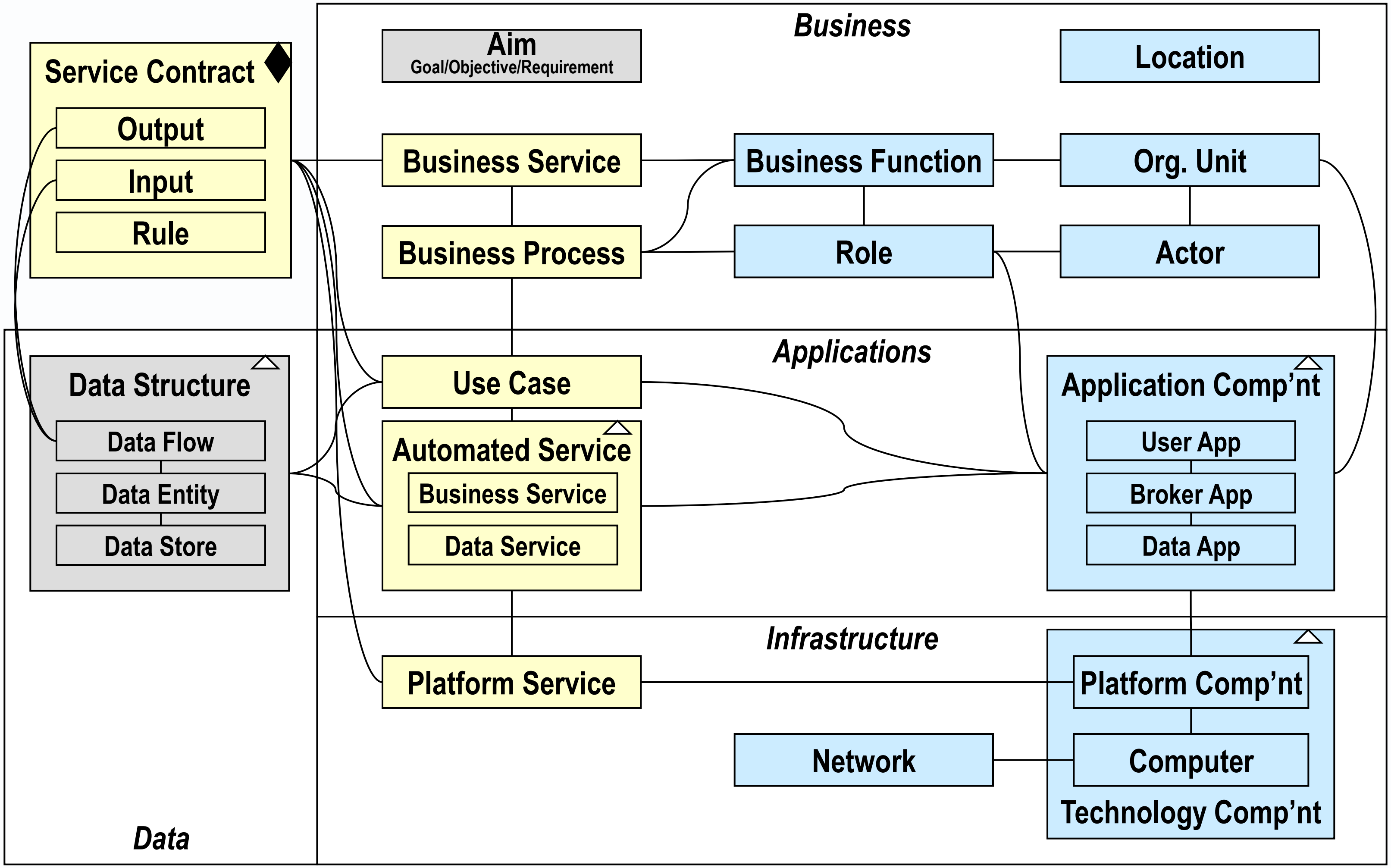


> “You can’t overstate the value of having a body of knowledge accessible in a central repository. In a split second, all stakeholders can find all the information they need in a consistent format, and they can view it in the way that enables them to do their job effectively. As a result, Dubai Customs has increased its agility and its ability to respond to new opportunities.”
>
> —Fadi Hindi, head of strategic IT planning and enterprise architecture, Dubai Customs

- From marketing of IBM’s Rational System Architect

Copyright Avancier Limited

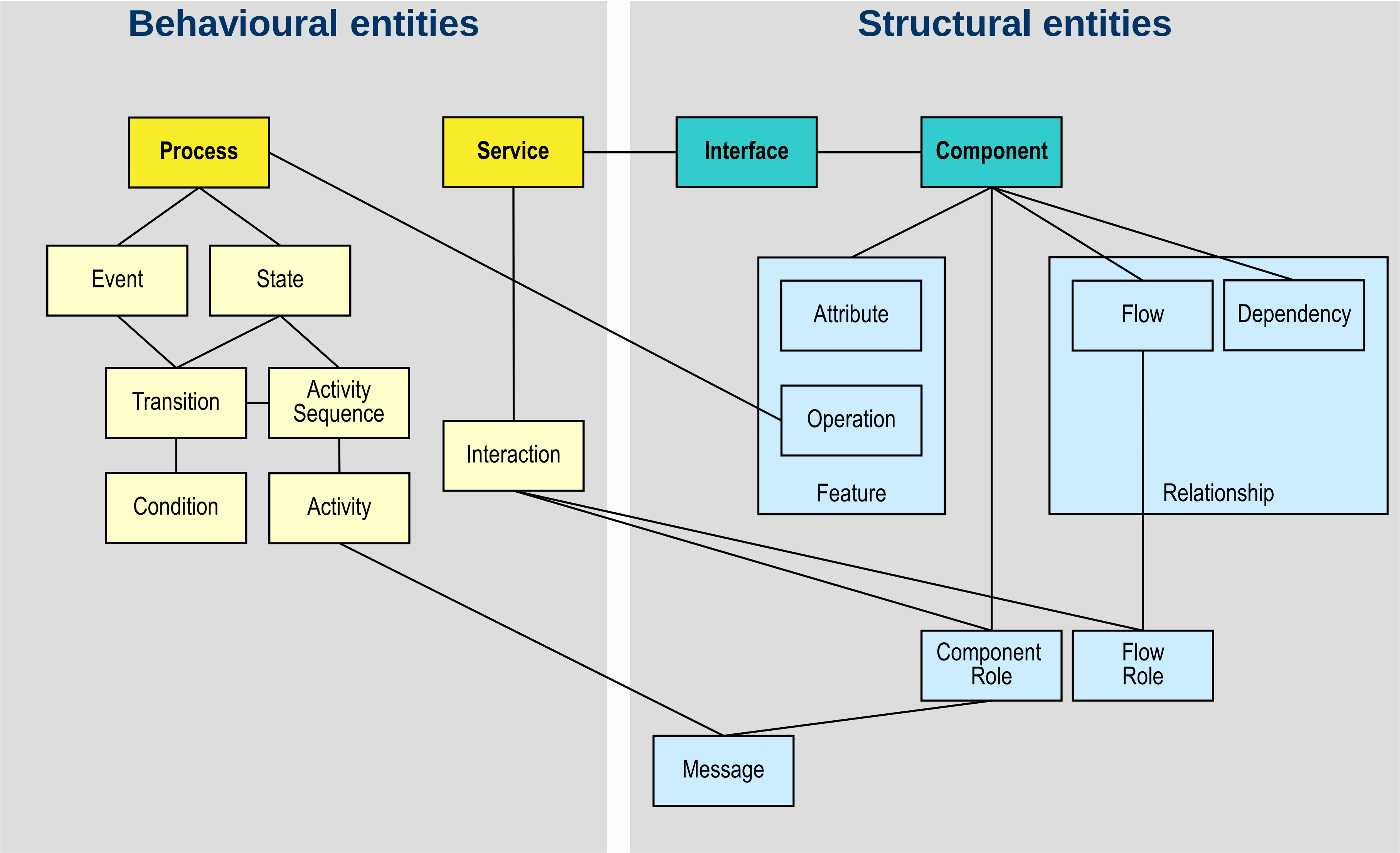
Agree the meta model of your architecture description



Copyright Avancier Limited

Other things you may want to show in diagrams

Avancier



Copyright Avancier Limited

For more guidance and techniques

- On the web site you can find scores of
  - Catalogues
  - Matrices
  - Diagrams

- No modelling language
  - Be it IDEF, UML or ArchiMate
  - Or this one - AML
- Is up to the job of describing everything that architects and designers want to describe

- And most industry standard notations grow through committee meetings until they are too complex to remember

Copyright Avancier Limited

AM: Diagrams

	Catalogues	Matrices	Diagrams
Architecture context			Value Chain/Stream diagram (T) Solution Concept/Vision diagram (T)
			Context diagram
Business architecture			Business Footprint diagram (T) Business Service/Information diagram (T) Functional Decomposition diagram (T) Goal/Objective/Service diagram (T) Product Lifecycle diagram (T) Business Use-Case diagram (T) Organization Decomposition diagram (T) Process Flow diagram (T) Event diagram (T)
			Business process map diagram Business goods and services flow diagram Location map diagram Location communication path diagram
Data architecture			Conceptual Data diagram (aka Business Data Model) (T) Logical Data diagram (T) Data Dissemination diagram (T) Data Lifecycle diagram (T) Data Security diagram (T) Data Migration diagram (T)
			Data Access Path (short-term process) diagram Data Flow Structure (regular expression) diagram
Application architecture Software architecture			Application Communication diagram (T) Application and User Location diagram (T) Application Use-Case diagram (T) Enterprise (System) Manageability diagram (T) Process/Application Realization diagram (T) Application Migration diagram (T) Software Engineering diagram (T) Software Distribution diagram (T)
			Software Layering table Component Dependency diagram Class diagram Sequence diagram
Infrastructure architecture			Environments and Locations diagram (T) Platform Decomposition diagram (T) Processing/Deployment diagram (T) Networked Computing/Hardware diagram (T) Communications Engineering diagram (T)

All these on the web site