# Avancier Methods (AM)
## PRODUCTS

Documentation framework

# Architects document baseline and target systems



**Contextual information**

stakeholders, concerns, requirements, principles, time, cost, and other precursors that architects must respond to.

**Architecture description**

Documentation

abstract level

inform

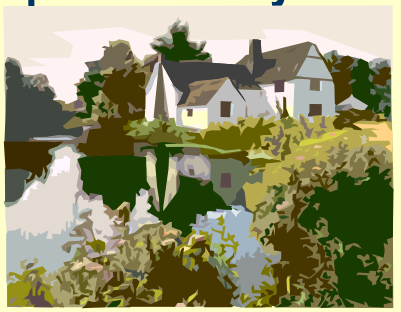
create and use

idealise

**Architects**

observe and envisage

**Operational system**



Baseline and Target

# Architecture documentation framework

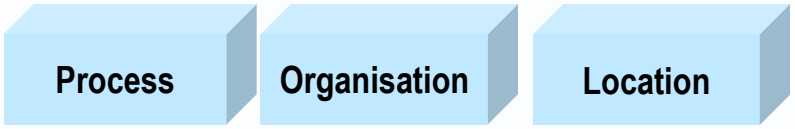
Avancier

- **Architectural building blocks - entities like** POLDAT
- **Architecture models/ 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, IDEF
- **Architecture deliverables**
  - Deliverables, work in progress or signed off.
  - Often contain architecture models/ artefacts
- **Architecture repository tools**
  - Records architectural entities, their attributes and inter-relationships
- **Pre-defined classifications and reference models**
  - Generalised taxonomies and common design patterns

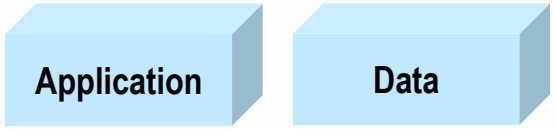
# Architectural entities - POLDAT

Avancier

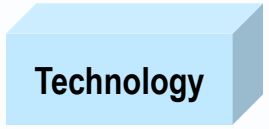
- CSC's "6 domains of change"
- Business



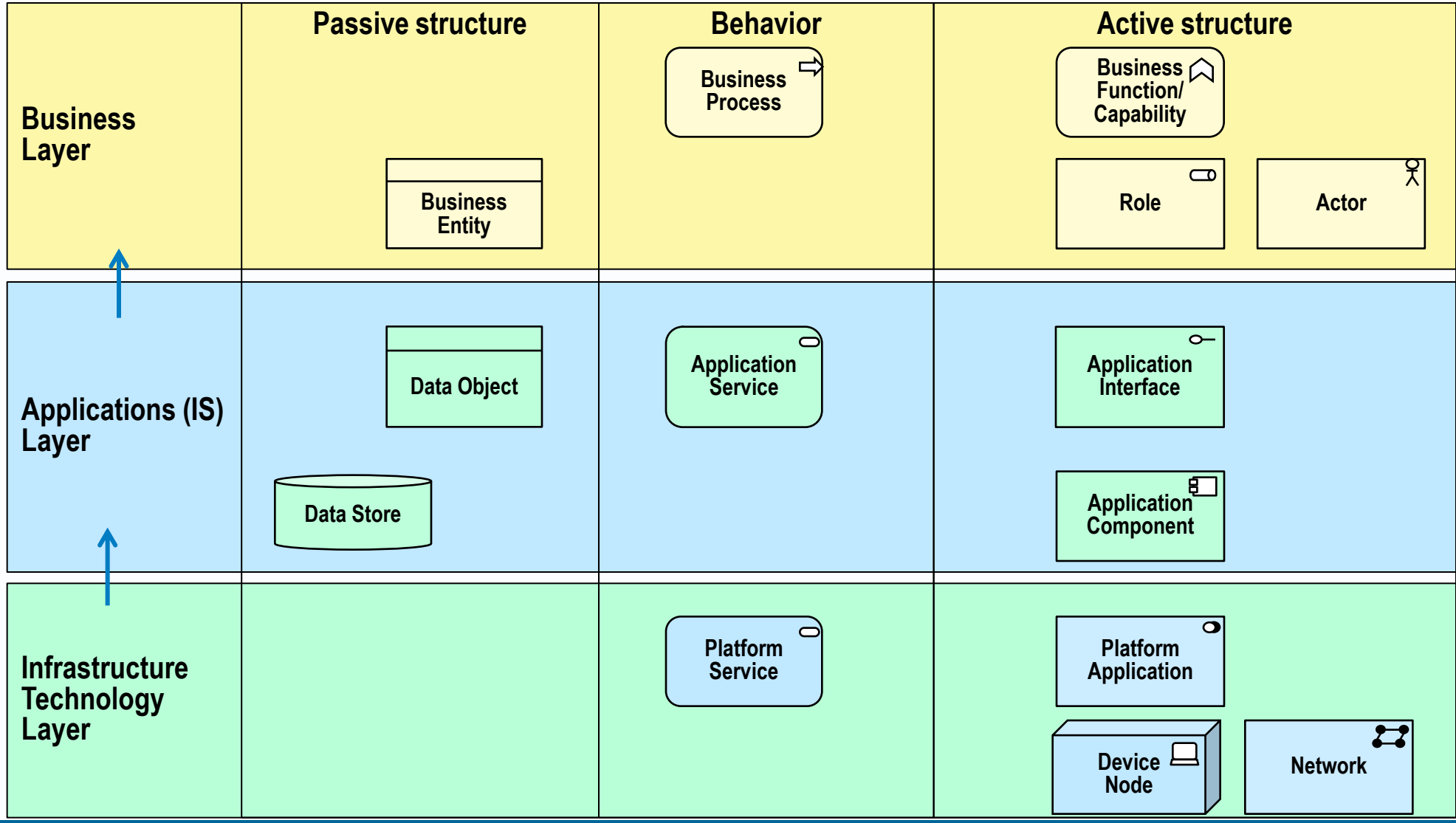

- Information Systems




- Infrastructure Technologies



# Architectural entities – commonly documented



# Architecture documentation framework

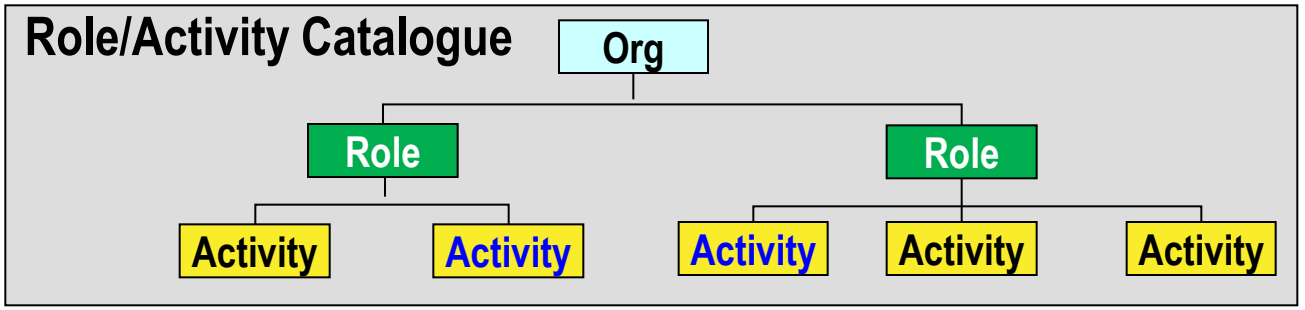
Avancier

- **Architectural building blocks - entities like** POLDAT
- **Architecture models/ 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, IDEF
- **Architecture deliverables**
  - Deliverables, work in progress or signed off.
  - Often contain architecture models/ artefacts
- **Architecture repository tools**
  - Records architectural entities, their attributes and inter-relationships
- **Pre-defined classifications and reference models**
  - Generalised taxonomies and common design patterns

# Three ways to relate architecture entities in “artefacts”



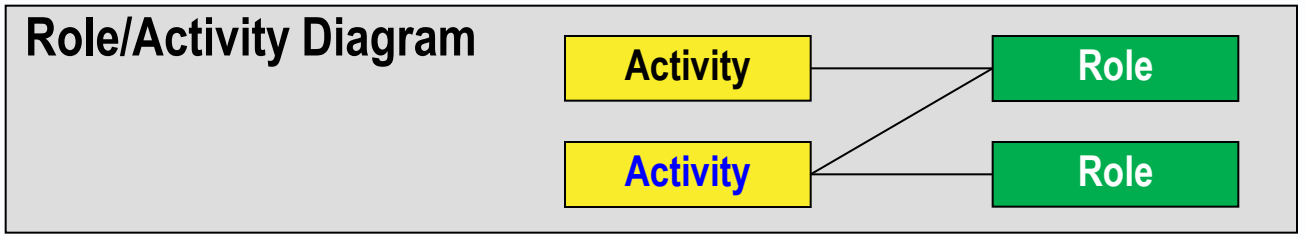
- A hierarchical catalogue describes one to many relationships well




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

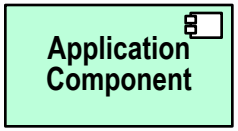
| Role/Activity Matrix | Role | Role |
|---|---|---|
| Activity | X | |
| Activity | X | X |

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



# An application portfolio catalogue

- Large enterprise have hundreds of applications
- (Few real catalogues record all the attributes in this template)

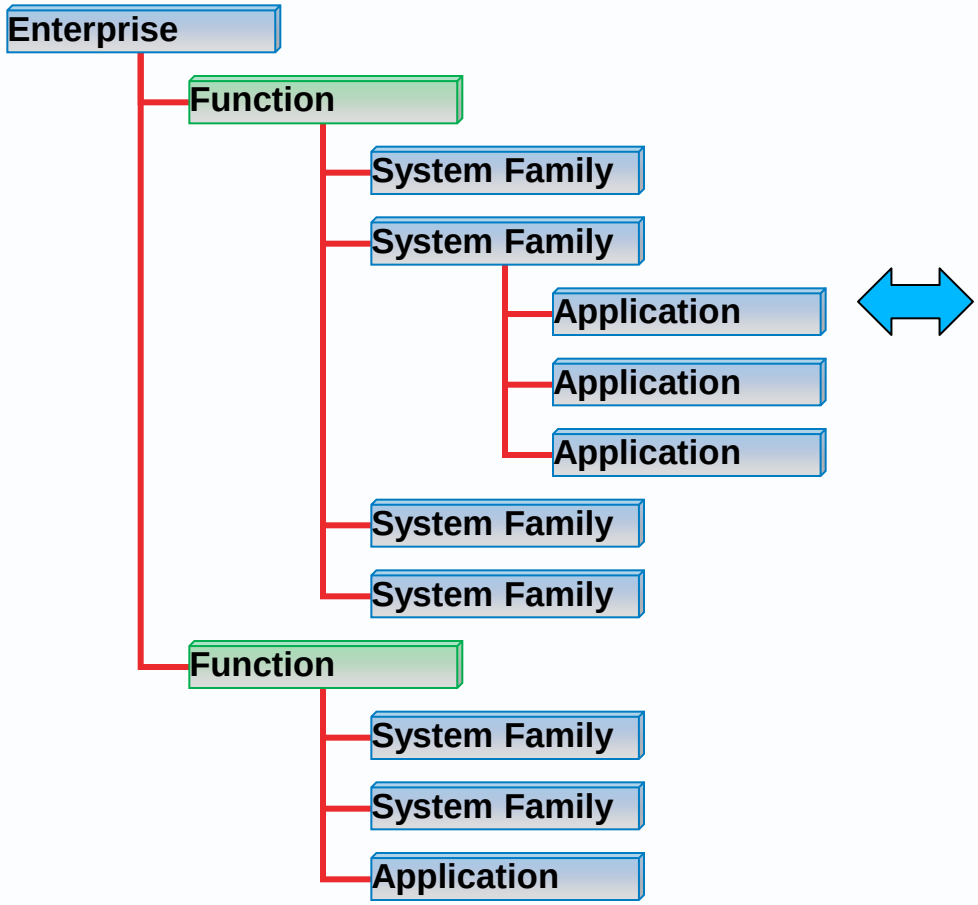



| **Application** | Primary key |
|---|---|
| **Application name** | |
| Cost to buy or build | |
| Cost to run and maintain per month | |
| Value to the enterprise | |
| Licence/contract expiry dates | |
| Status | |
| Class | |
| Roles (owners, users, maintainers) | |
| Business functions/capabilities supported | |
| Organisation units supported | |
| Applications/components communicated with | |
| Data stores accessed | |
| Networks used | |
| Hardware/software platform technologies | |
| Etc. | |

# Mapping applications to a functional decomposition

Avancier

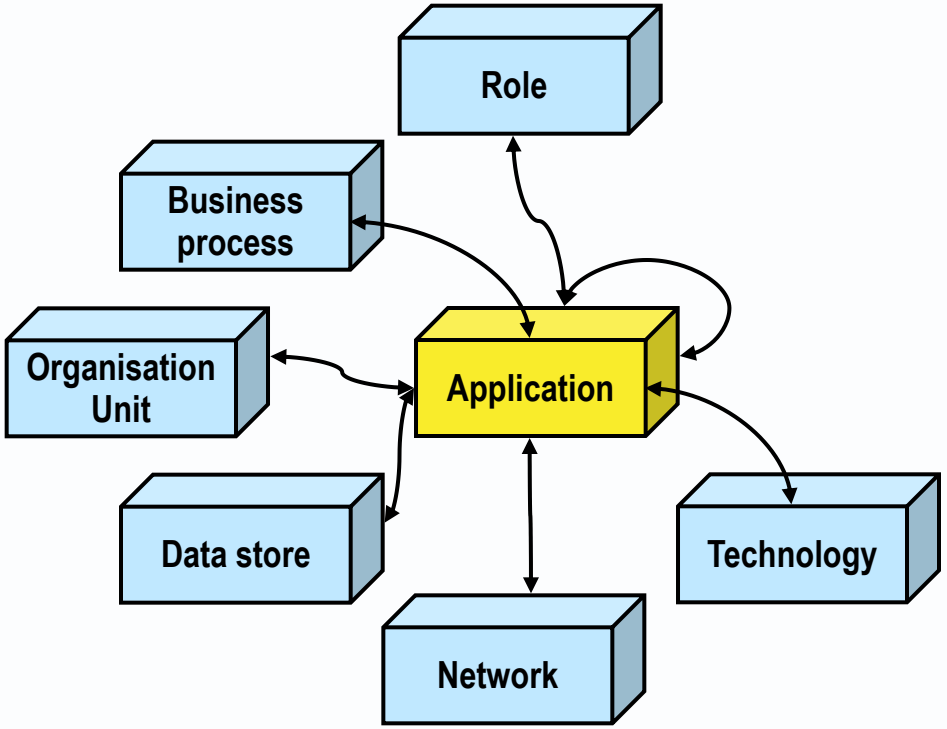
Group application and assign them to high-level business functions




| **Application** |
| --- |
| **Application name** |
| Cost to buy or build |
| Cost to run and maintain per month |
| Value to the enterprise |
| Licence/contract expiry dates |
| Status |
| Class |
| Roles (owners, users, maintainers) |
| Business functions/capabilities supported |
| Organisation units supported |
| Applications/components communicated with |
| Data stores accessed |
| Networks used |
| Hardware/software platform technologies |
| Etc. |

# Some attributes imply relationships to other entities

- Some attributes are “foreign keys”, the primary keys of other Entities



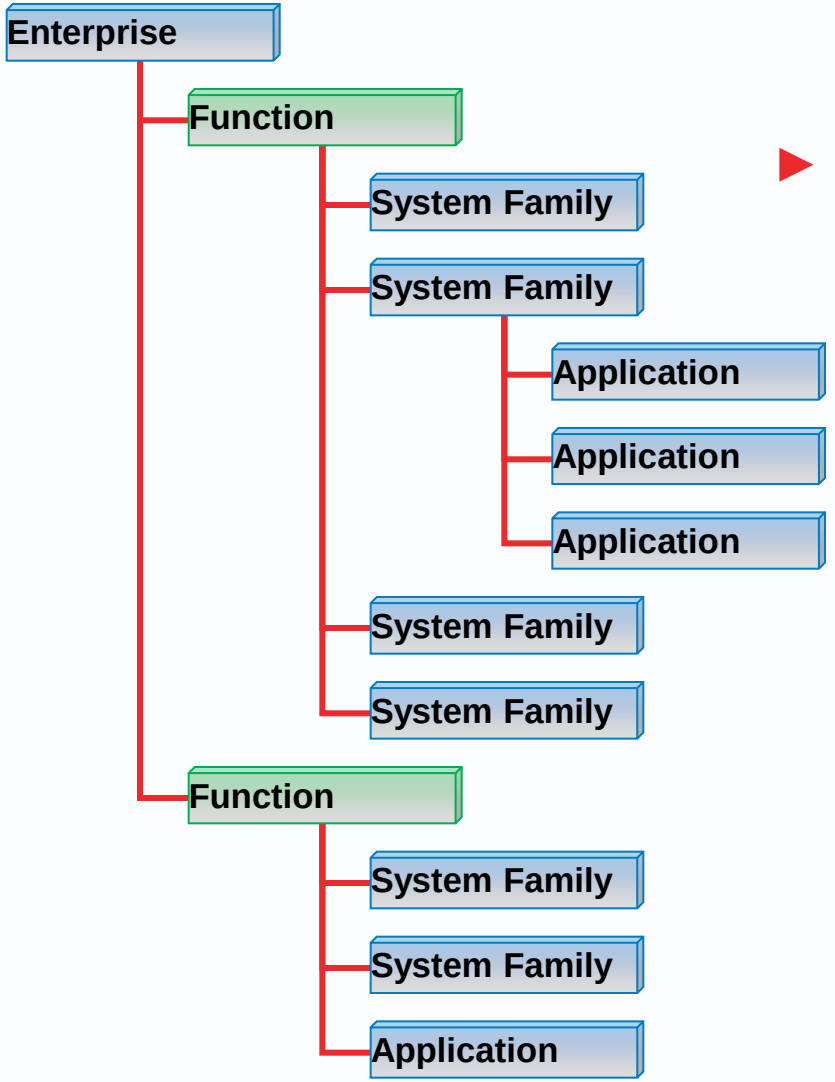

| **Application** |
| --- |
| **Application name** |
| Cost to buy or build |
| Cost to run and maintain per month |
| Value to the enterprise |
| Licence/contract expiry dates |
| Status |
| Class |
| Roles (owners, users, maintainers) |
| Business functions/capabilities supported |
| Organisation units supported |
| Applications/components communicated with |
| Data stores accessed |
| Networks used |
| Hardware/software platform technologies |
| Etc. |

Primary key

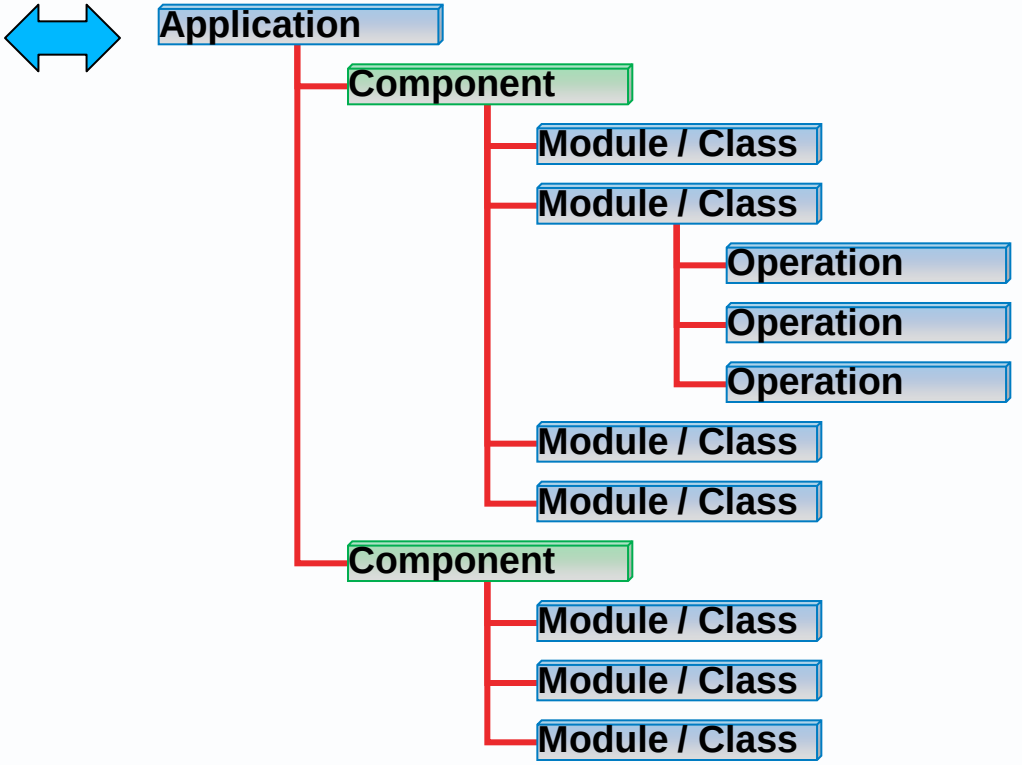
Foreign keys

# Deep structures are decoupled into distinct catalogues






- In practice, to maintain a complete top-to-bottom hierarchy seems impractical



# AM: Catalogues

- Driver/Goal/Objective catalogue (T)
- Directive (principle/policy) catalogue (T)
- Stakeholder catalogue/matrix (T)
- Technology Standards catalogue (T)
- Driver or concern catalogue
- Aim (goal/objective/requirement) catalogue
- Regulations catalogue

- Organization/Role/Actor catalogue (T)
- Role catalogue (T)
- Business Service/Function catalogue (T)
- Contract/Measure catalogue (T)
- Process/Event/Control/Product catalogue (T)
- Location catalogue (T)
- Product, goods or service catalogue
- Service Level Agreement (service catalogue)
- Business term or rule catalogue
- Actor catalogue
- Location type catalogue

- Data Entity/Data Component catalogue (T)
- Data Component catalogue
- Data Entity catalogue
- Canonical data model: item catalogue
- Application Portfolio catalogue (T)
- Interface catalogue (T)
- Data Flow catalogue

- UIs and API/service catalogue
- Application use case catalogue
- Service catalogue
- Qualities of a Service
- Component catalogue

- Platform Services catalogue (TRM)
- Technology Standards catalogue (T)
- Technology Portfolio catalogue (T)
- Client Device catalogue
- Server Device catalogue
- Channel catalogue

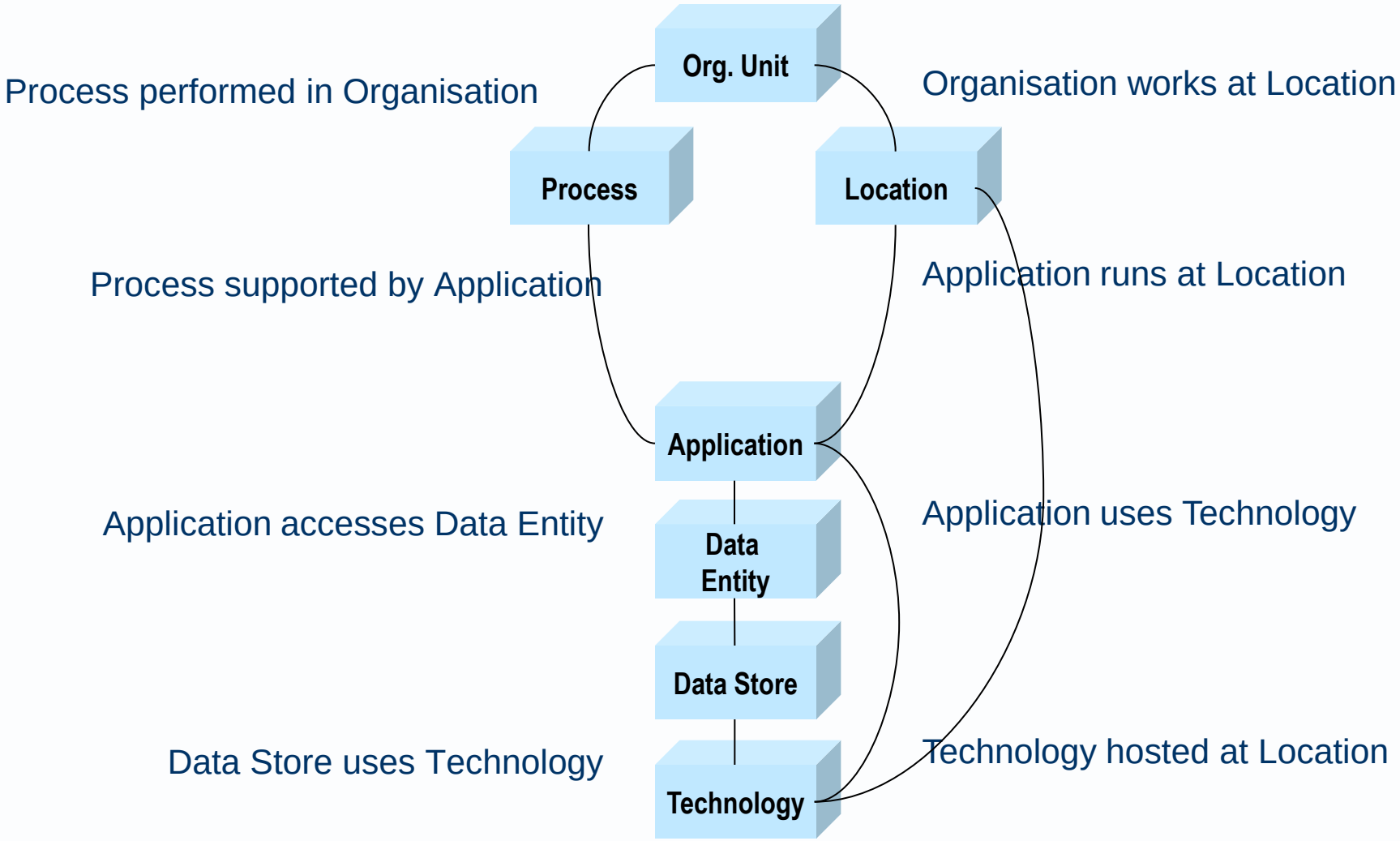
You can find all these on the web site

# Architecture documentation framework

Avancier

- **Architectural building blocks - entities like** POLDAT
- **Architecture models/ 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, IDEF
- **Architecture deliverables**
  - Deliverables, work in progress or signed off.
  - Often contain architecture models/ artefacts
- **Architecture repository tools**
  - Records architectural entities, their attributes and inter-relationships
- **Pre-defined classifications and reference models**
  - Generalised taxonomies and common design patterns

# Relating architectural entities (POLDAT after CSC)



# Inter-entity relationships as tables, grids or matrices

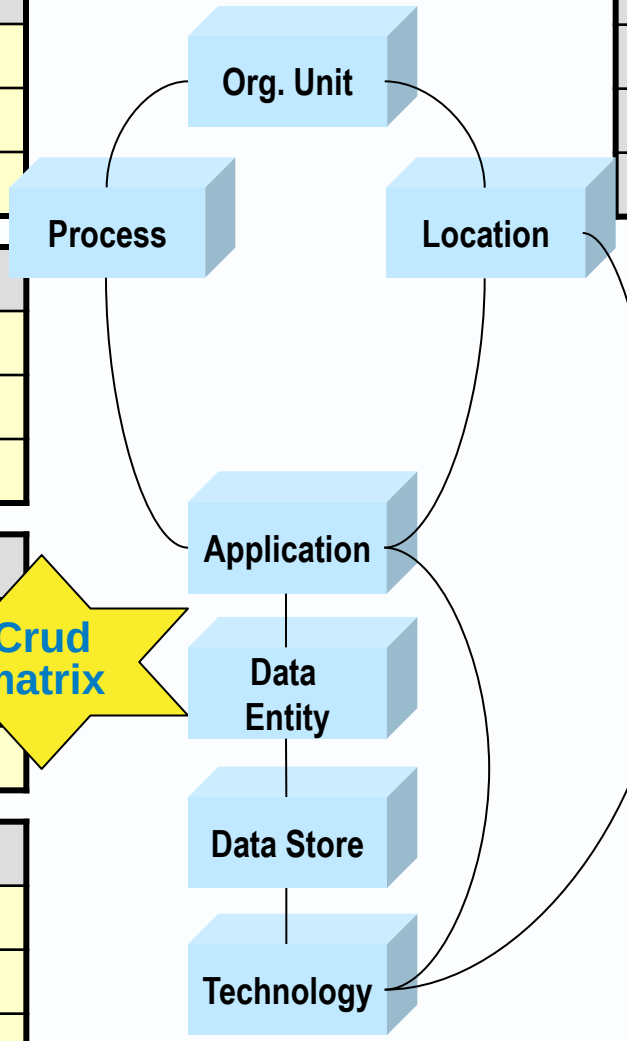
Avancier

| | Process | Process | Process |
|---|---|---|---|
| Org | Performs | | |
| Org | | Performs | Performs |
| Org | | | Performs |

| | Process | Process | Process |
|---|---|---|---|
| App | supports | | |
| App | | supports | supports |
| App | | | supports |

| | App | App | App |
|---|---|---|---|
| Entity | cru | rud | |
| Product | crud | ru | r |
| Asset | r | crud | r |

| | Tech | Tech | Tech |
|---|---|---|---|
| Data store | uses | | |
| Data store | | uses | uses |
| Data store | | | |




| | Loc | Loc | Loc |
|---|---|---|---|
| Org | Works at | | |
| Org | | Works at | Works at |
| Org | Works at | Works at | |

Data Dissem. Matrix

| | Data Store | Data Store | Data Store |
|---|---|---|---|
| Entity | M | C | |
| Product | | M | C |
| Asset | | M | C |

| | Tech | Tech | Tech |
|---|---|---|---|
| App | uses | | |
| App | | uses | uses |
| App | | | |

# AM: Matrices

- Goal or requirements traceability matrix
- Business Interaction matrix (T)
- Actor/Role matrix (T)
- Organisation/Location matrix
- Organisation/Activity matrix
- Business process/Use case matrix
- Data Entity/Business function matrix (T)
- Application/Data matrix (T)
- Data dissemination matrix
- Application/Organization matrix (T)
- Role/Application matrix (T)
- Application/Function or Activity matrix (T)
- Application Interaction matrix (T)
- Application/Technology matrix (T)
- Data component/Technology matrix

You can find all these on the web site

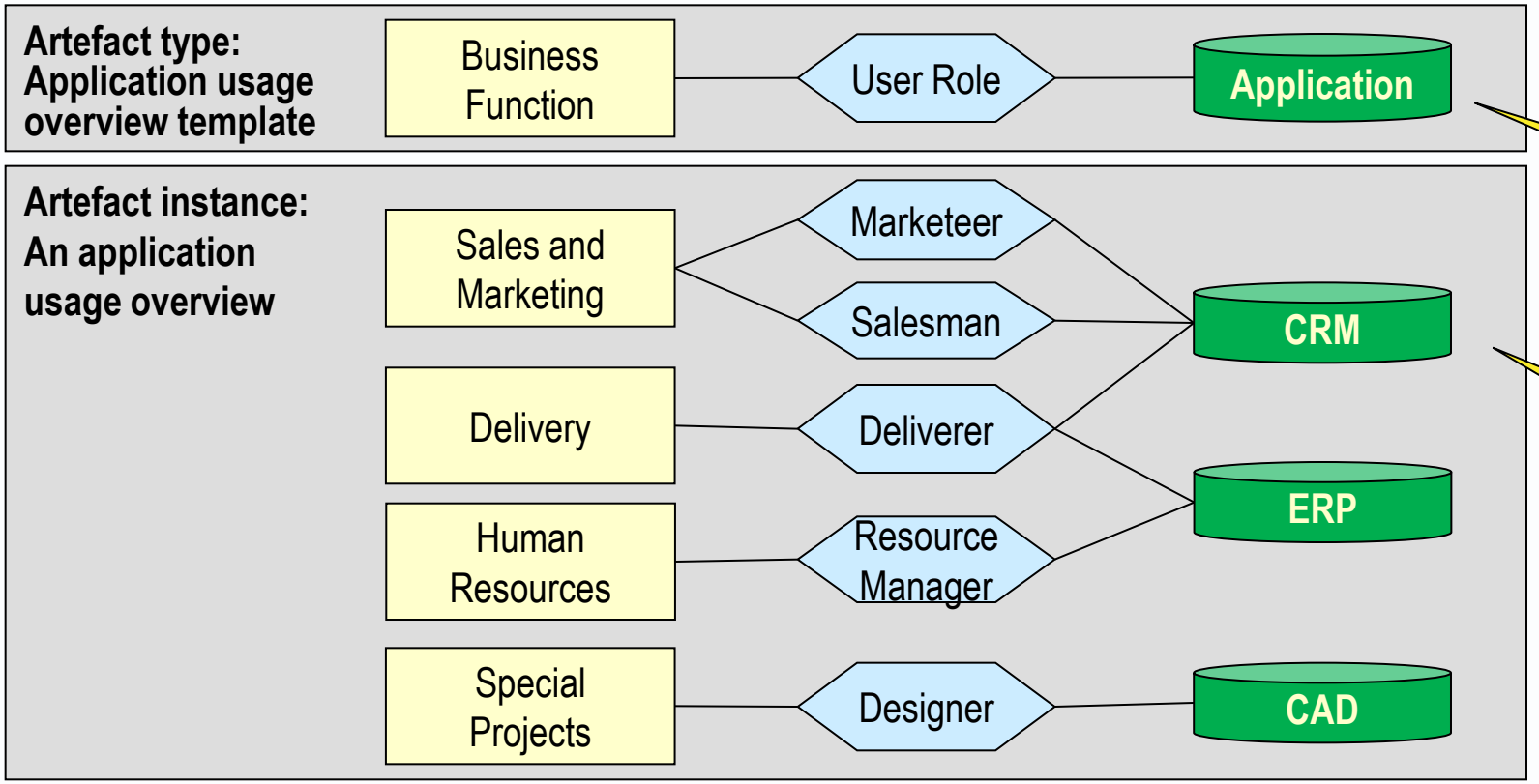
# Architecture documentation framework

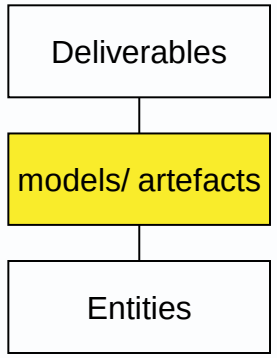
Avancier

- **Architectural building blocks - entities like** POLDAT
- **Architecture models/ 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, IDEF
- **Architecture deliverables**
  - Deliverables, work in progress or signed off.
  - Often contain architecture models/ artefacts
- **Architecture repository tools**
  - Records architectural entities, their attributes and inter-relationships
- **Pre-defined classifications and reference models**
  - Generalised taxonomies and common design patterns

# Relating architectural entities in a diagram

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



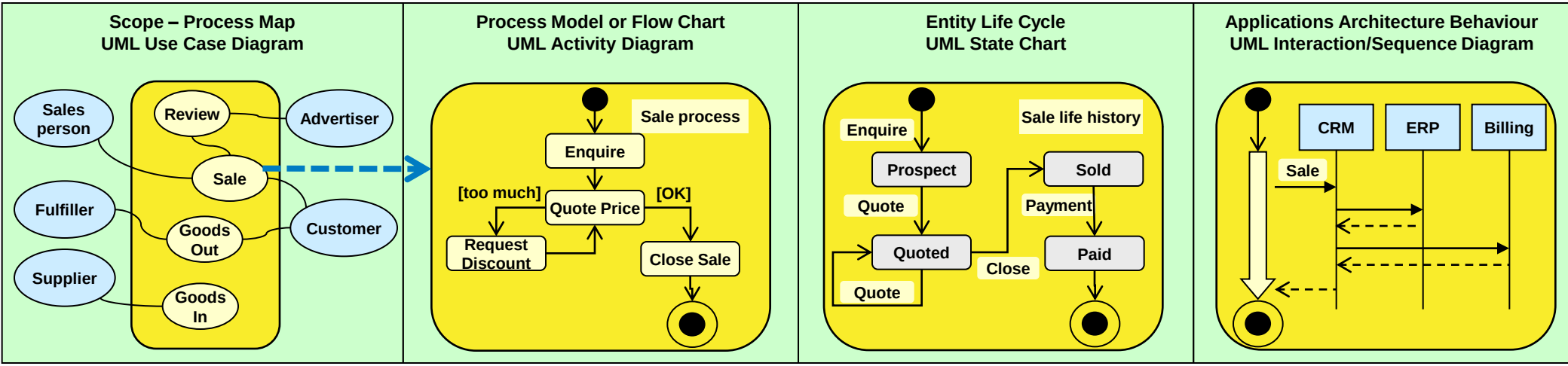




**View point**

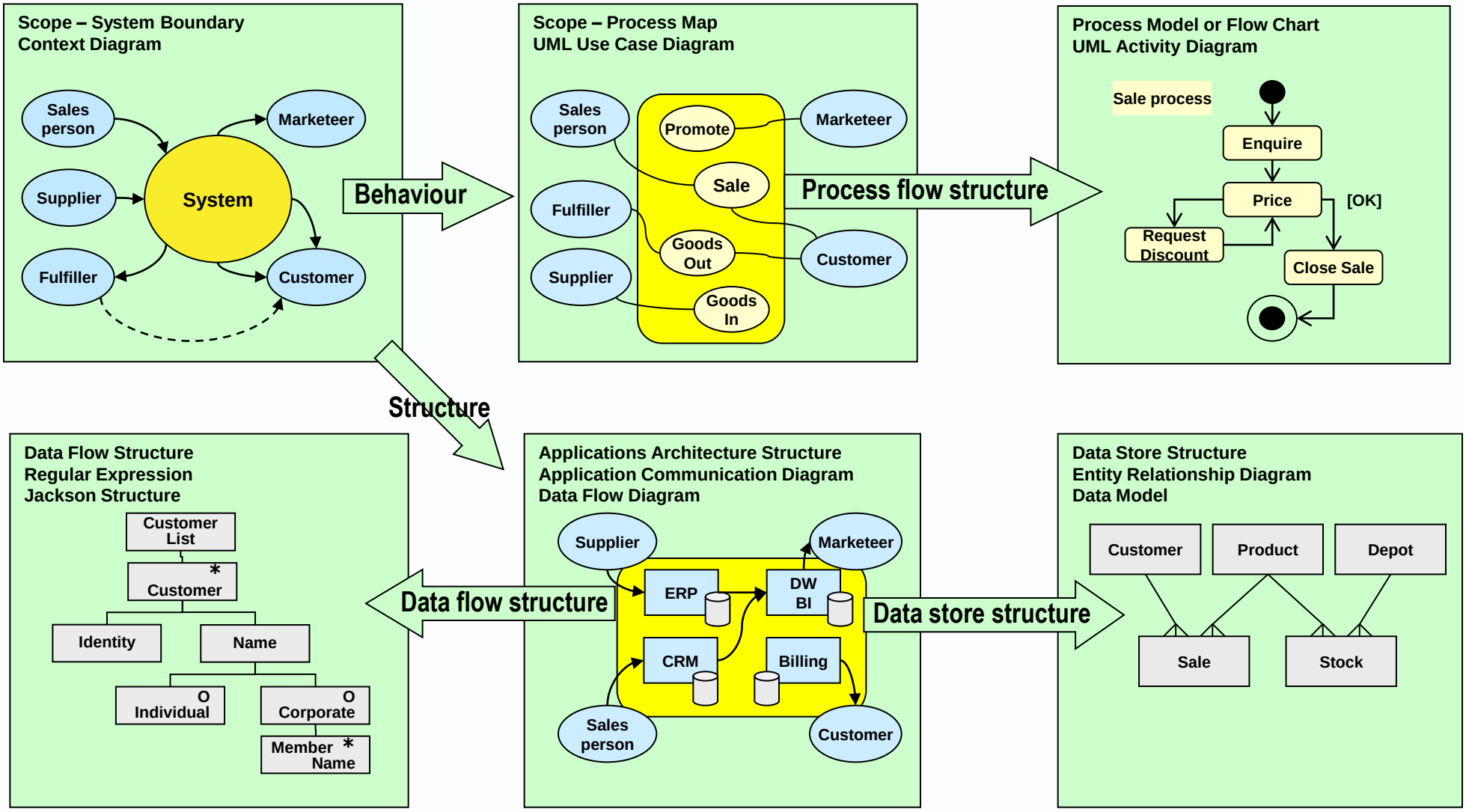
**View**

# Diagrams that show behavioural views of system

- A process has a time dimension; it runs from start to end

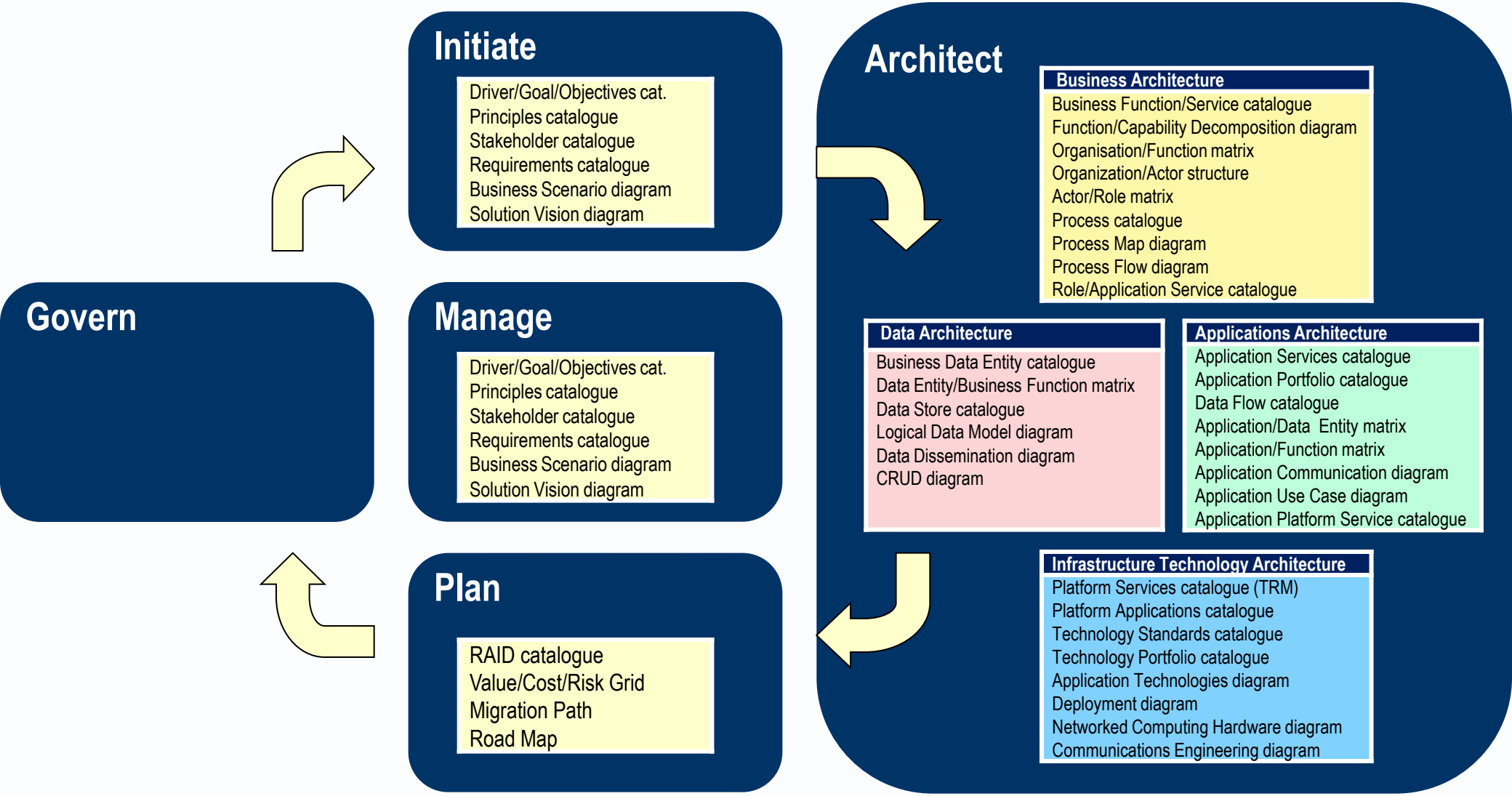


# Diagrams form inter-related views of an architecture



# Architecture Artefacts mapped to the AM process

Avancier

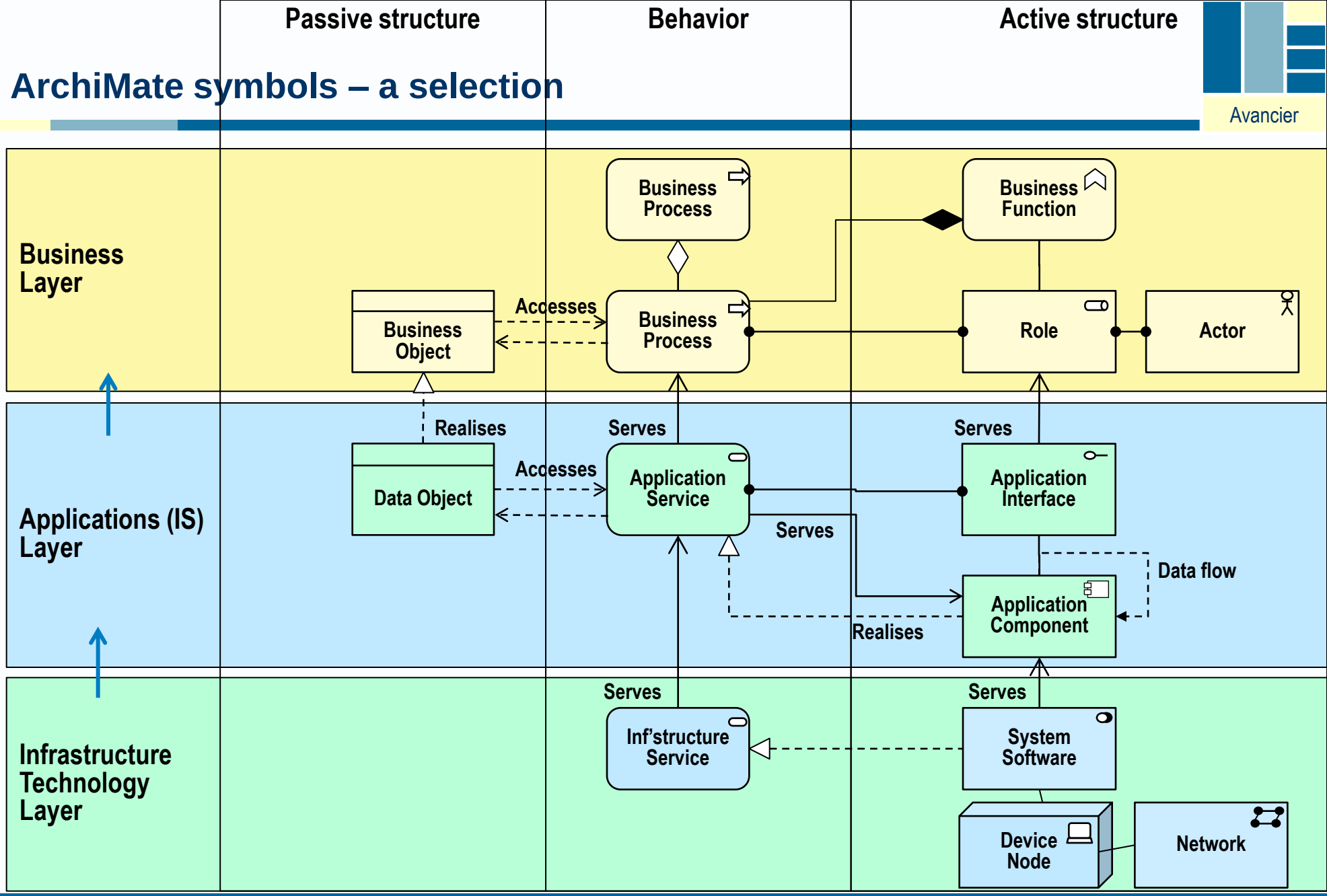


# Architecture documentation framework

Avancier

- **Architectural building blocks - entities like** POLDAT
- **Architecture models/ 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, IDEF
- **Architecture deliverables**
  - Deliverables, work in progress or signed off.
  - Often contain architecture models/ artefacts
- **Architecture repository tools**
  - Records architectural entities, their attributes and inter-relationships
- **Pre-defined classifications and reference models**
  - Generalised taxonomies and common design patterns

# ArchiMate symbols – a selection



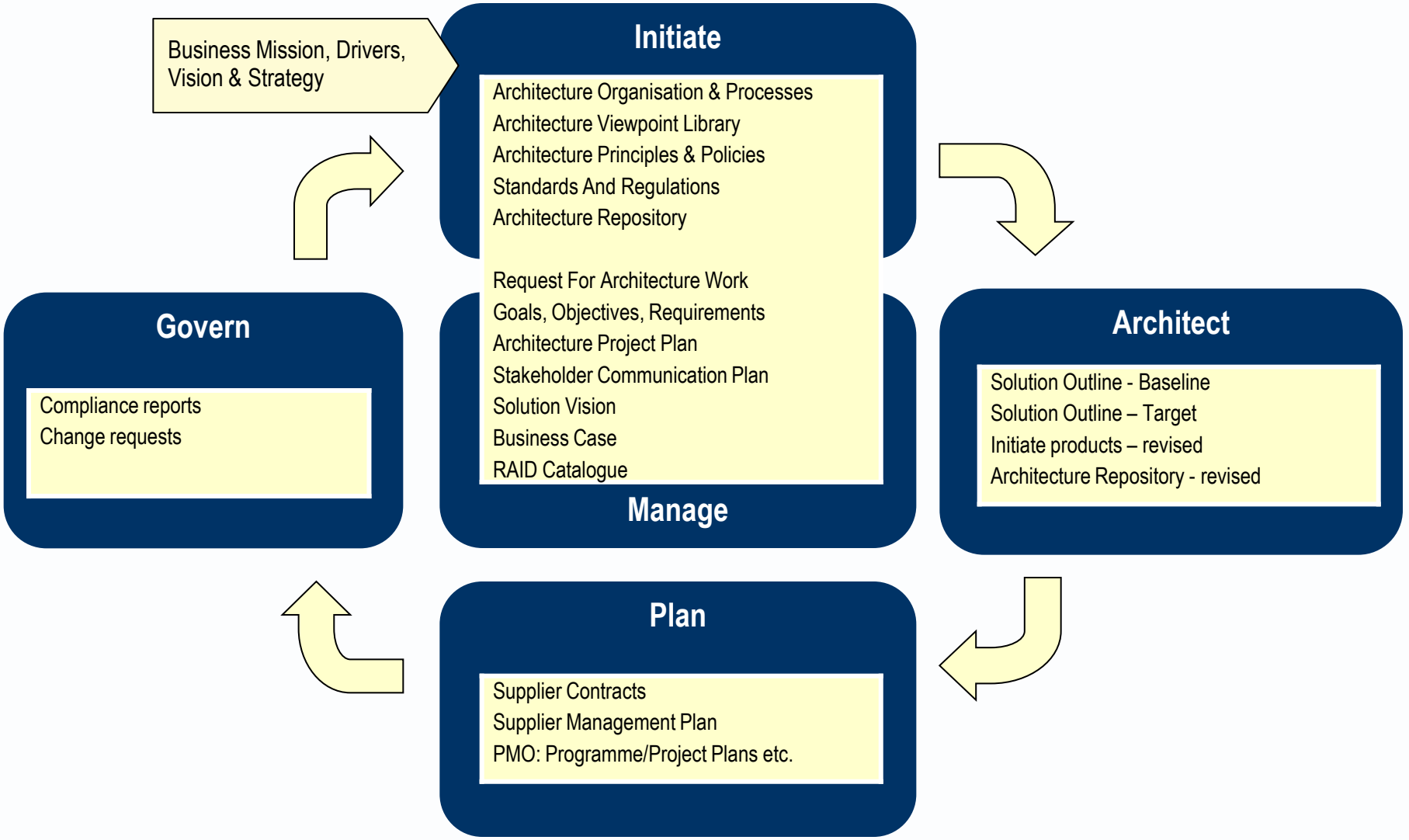
# Architecture documentation framework

Avancier

- **Architectural building blocks - entities like** POLDAT
- **Architecture models/ 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, IDEF
- **Architecture deliverables**
  - Deliverables, work in progress or signed off.
  - Often contain architecture models/ artefacts
- **Architecture repository tools**
  - Records architectural entities, their attributes and inter-relationships
- **Pre-defined classifications and reference models**
  - Generalised taxonomies and common design patterns

# Architecture Deliverables mapped to the AM process

Avancier



# Architecture documentation framework

Avancier

- **Architectural building blocks - entities like** POLDAT
- **Architecture models/ 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, IDEF
- **Architecture deliverables**
  - Deliverables, work in progress or signed off.
  - Often contain architecture models/ artefacts
- **Architecture repository tools**
  - Records architectural entities, their attributes and inter-relationships
- **Pre-defined classifications and reference models**
  - Generalised taxonomies and common design patterns

# How architecture description develops

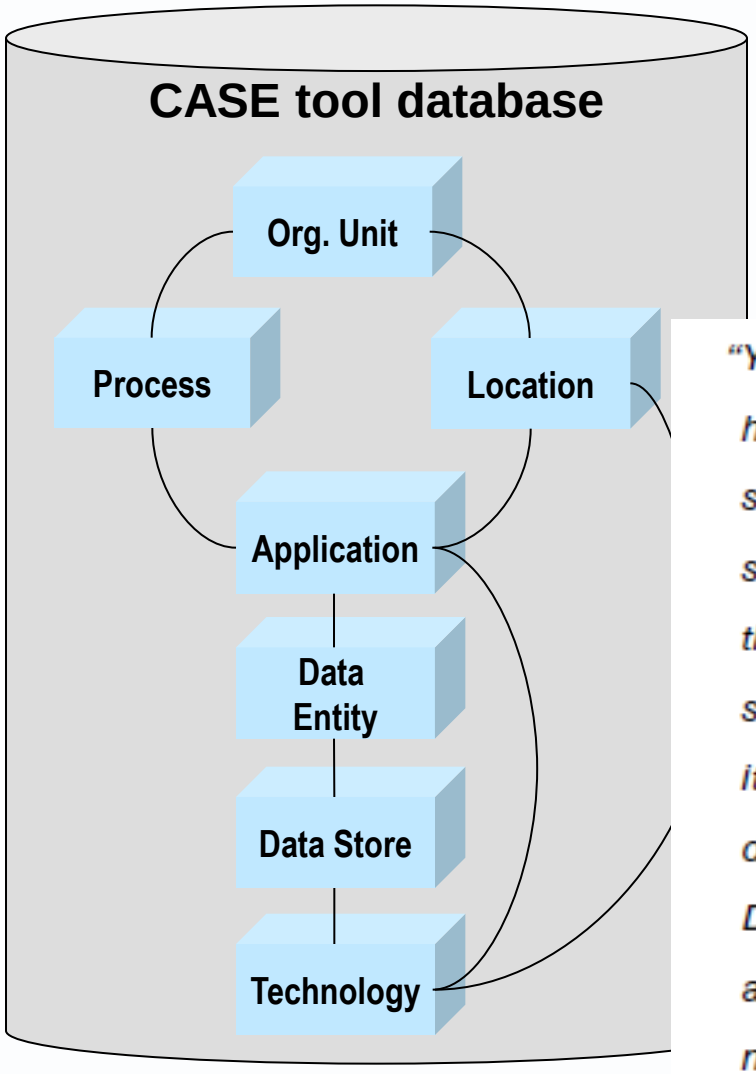


- You start off writing a deliverable as one document
- You write sections covering business, apps and technology concerns, mentioning entities such human roles and processes, apps and technologies
- You insert various artefacts (tables and diagrams) to show the relationships between the Entities
- The artefacts refer to the entities by name
- You describe the entities more fully in catalogues in appendices
- You divide the document into documents for different stakeholders with different concerns
- Your overall description has now become so complex and distributed that you turn the appendices into a set of spreadsheets (a repository) from which you copy content into deliverables for stakeholders to read.

# Architecture descriptions can be stored in a repository



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



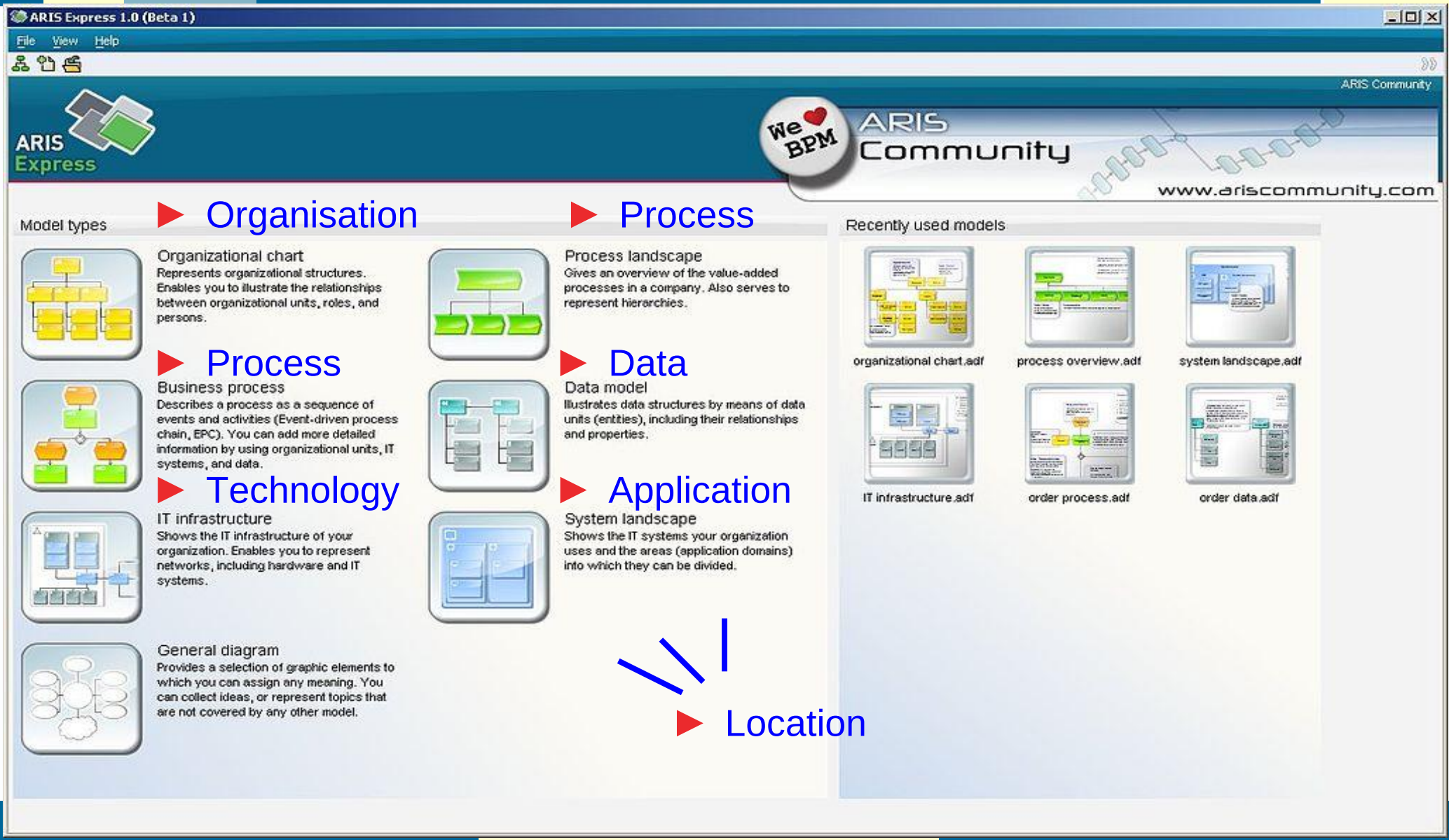

- From marketing of IBM's Rational System Architect

*"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

# POLDAT entities documented using ARIS tool

Avancier



# There is no best CASE tool for Enterprise and Solution Architecture



- Different tools are strong in different areas.

# From “The best EA tool?” at avancier.website for discussion



| TOOL | COMPANY | Typical class comment |
|---|---|---|
| ABACUS | Avolution | |
| Alfabet | Software AG | EA and Portfolio management |
| Archi | http://archi.cetis.ac.uk/ | Free |
| Architect | Bizzdesign | Friendly to ArchiMate and TOGAF |
| ARIS | Software AG | Process model oriented |
| Casewise | Casewise | Clunky (rumour of integration with ERwin) |
| Enterprise Architect | Sparx Systems | Cheap, adequate at project level |
| ERwin | CA Technologies | Data modelling (rumour of integration with Casewise) |
| MagicDraw | No Magic | UML modeller - popular with Java developers |
| Mega | Mega | Multi-faceted upper CASE tool |
| MooD Transformation Technology | MooD | Management-oriented upper CASE tool |
| Navigate, Insight and Architect, | Troux Technologies | Multi-faceted upper CASE tool |
| ProVision | Open Text (ex Metastorm) | Process model oriented |
| Rational software architect | IBM | Developer-oriented lower CASE tool |
| Rational system architect | IBM | Clunky upper CASE tool |
| Visio / I-Server | Micosoft / Orbus Software | Good for Visio users |
| Visual Paradigm | Visual Paradigm | Data models and UML, imports Visio |

# Architecture documentation framework



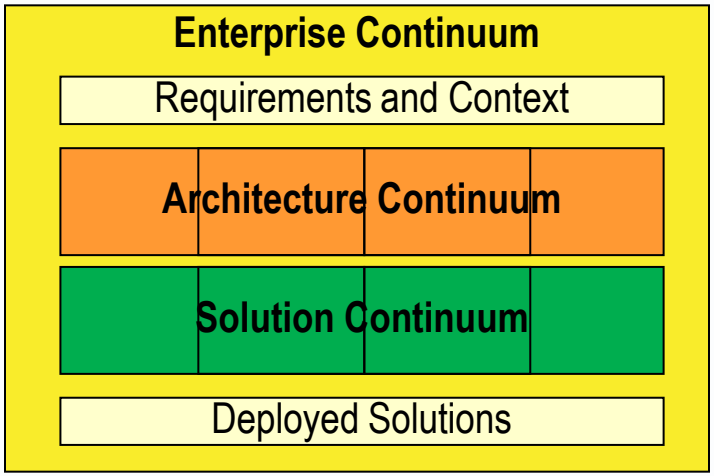
- **Architectural building blocks - entities like** POLDAT
- **Architecture models/ artefacts**
  - **Catalogues** that list architectural entities with attributes
  - **Matrices** that relate architecture entities
  - **Diagrams** that describe and/or relate architectural entities
- **Architecture models and languages** e.g. ArchiMate, UML, IDEF
- **Architecture deliverables**
  - Deliverables, work in progress or signed off.
  - Often contain architecture models/ artefacts
- **Architecture repository tools**
  - Records architectural entities, their attributes and inter-relationships
- **Pre-defined classifications and reference models**
  - Generalised taxonomies and common design patterns

# Architecture artefact classification schemes



- Ways of classifying architectural documentation so you file and retreive elements of it
- Nothing more or less than pigeon holes for artefacts (catalogues, matrices, diagrams)

## TOGAF's Enterprise Continuum




## Zachman Framework

| | Columns - questions | | | | | |
|---|---|---|---|---|---|---|
| **Rows - reification** | What | How | Where | Who | When | Why |
| Scope Contexts | | | | | | |
| Business Concepts | | | | | | |
| System Logic | | | | | | |
| Technology Physics | | | | | | |
| Component assemblies | | | | | | |
| Operations Instance classes | | | | | | |

# 1st classification - TOGAF's "Enterprise Continuum"

Avancier

- Rows = levels of idealisation
- Columns = levels of generalisation.

| Idealisation \ Generalisation | Foundation (Universal) | Common Systems (Fairly generic) | Industry (Fairly specific) | Organisation (Uniquely configured) |
|---|---|---|---|---|
| Requirements and Context | | | | |
| Architecture Continuum (Logical Models) | Foundation Architecture | Common System Architecture | Industry Architecture | Organisation Architecture |
| Solution Continuum (Physical Models) | Foundation Solutions | Common System Solutions | Industry Solutions | Organisation Solutions |
| Deployed solutions | | | | |

Ideal

Real

- Architects can assign each description artefact to a cell of the schema, then use the schema as an index to find models/ artefacts in a repository.

# 2nd classification - Zachman framework

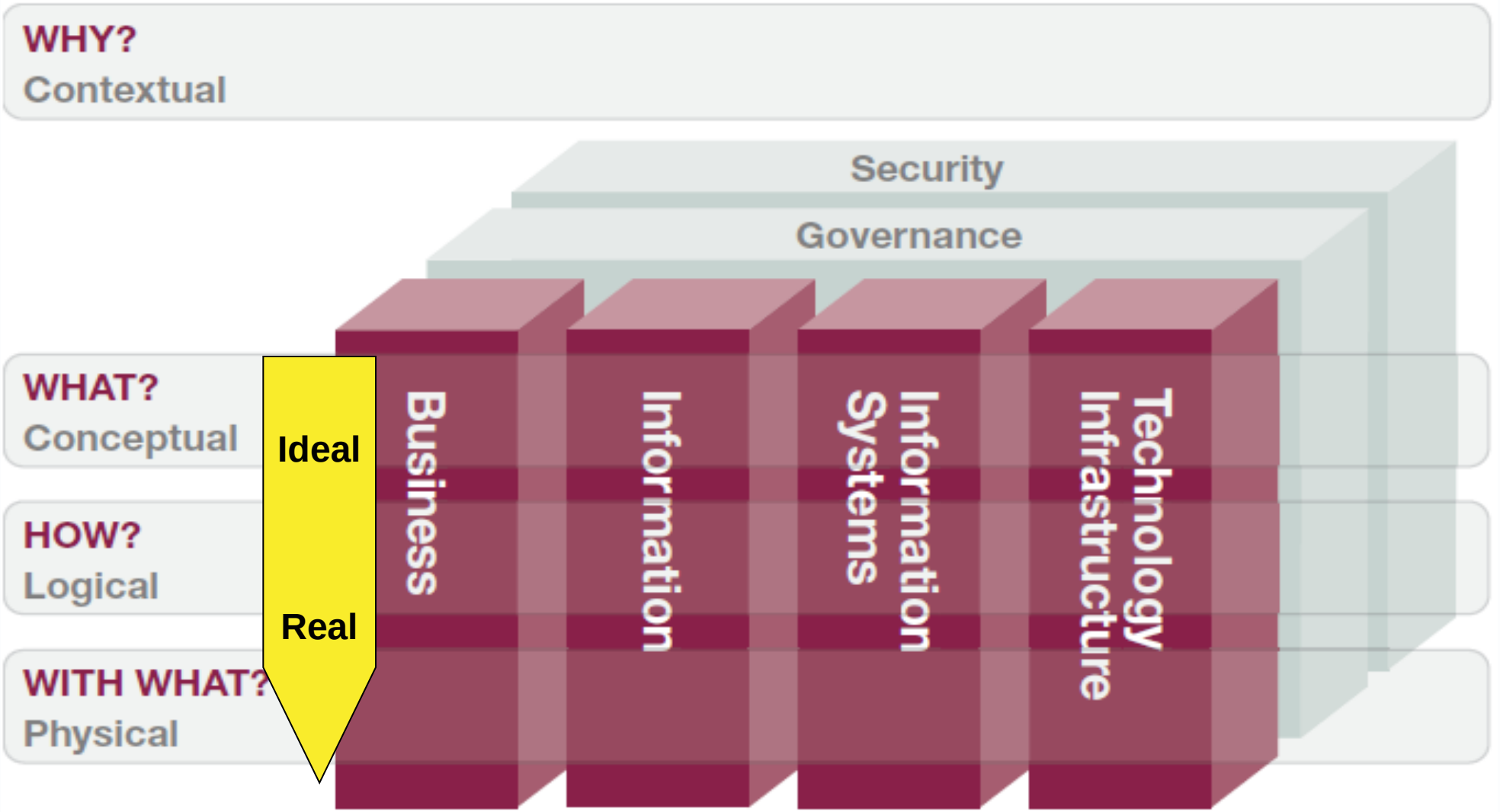
Avancier

| Zachman Framework v3 | | What | How | Where | Who | When | Why |
|---|---|---|---|---|---|---|---|
| Idealisation | Stakeholder perspective | Inventory sets | Process flows | Distribution networks | Responsibility assignments | Timing cycles | Motivation intentions |
| Scope Contexts | Executive | List inventory types | List process types | List distribution types | List responsibility types | List timing types | List motivation types |
| Business Concepts | Business anagement | Business entities & relationships | Business & input output | Business location & connection | Business role & work product | Business interval & moment | Business ends & means |
| System Logic | Architect | System entities & relationships | System & input output | System location & connection | System role & work product | System interval & moment | System ends & means |
| Technology Physics | Engineer | Technology entities & relationships | Technology input & output | Technology & location connection | Technology role & work product | Technology interval & moment | Technology ends & means |
| Tool components | Technician | Tool entities & relationships | Tool input & output | Tool location & connection | Tool role & work product | Tool interval & moment | Tool ends & means |
| Operations - Instance classes | Enterprise | Operations entities & relationships | Operations entities & relationships | Operations entities & relationships | Operations entities & relationships | Operations entities & relationships | Operations entities & relationships |

Ideal

Real

# 3rd classification - Cap Gemini's IAF



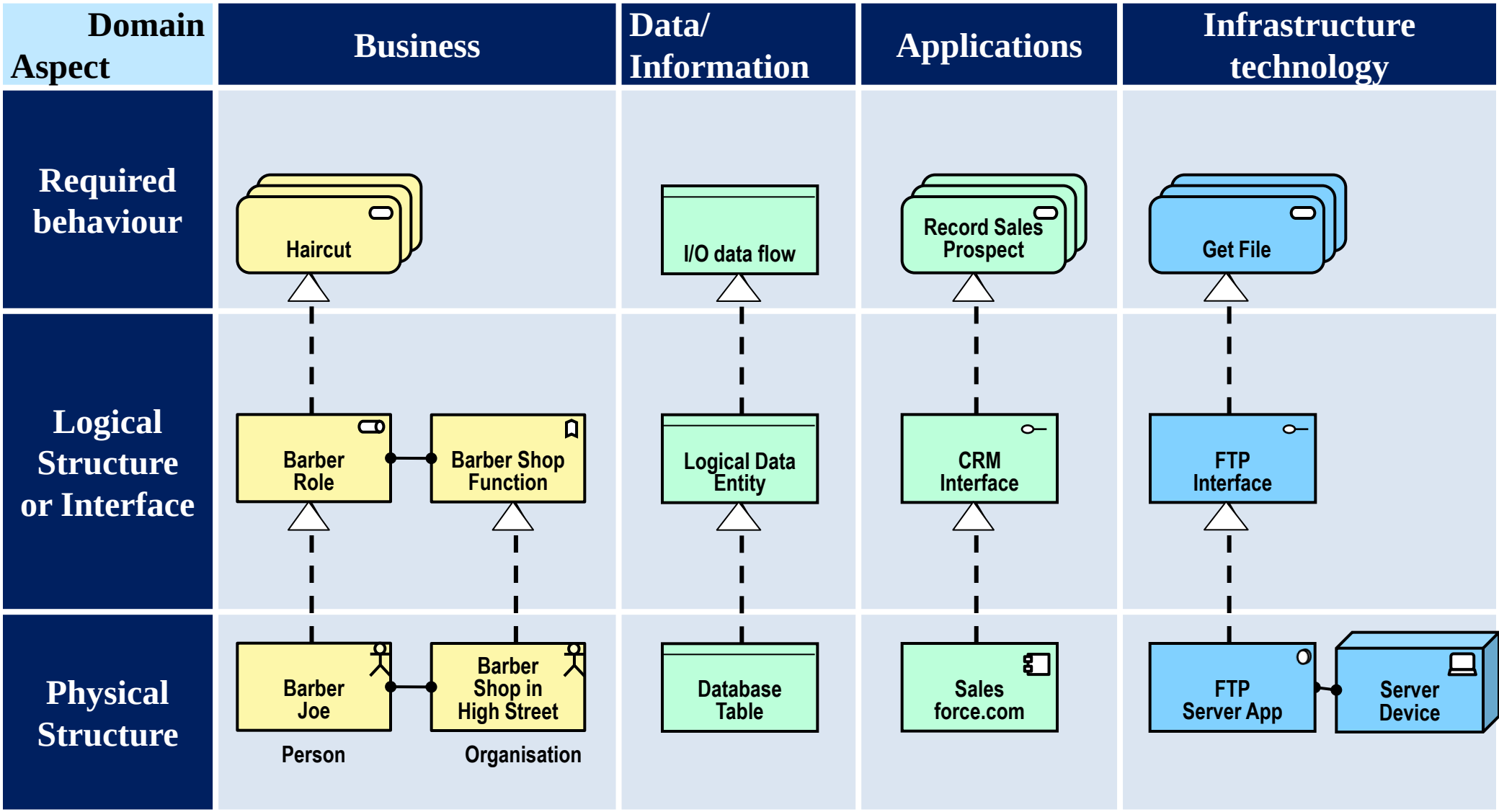
# AM classification by Architecture Domain and Aspect

Avancier

| Domain / Aspect | Business | Data/ Information | Applications | Infrastructure technology |
|---|---|---|---|---|
| **Required behaviour** | Business service<br>Business process | I/O Data flow | Application service | Platform service |
| **Logical Structure or Interface** | Business function<br>Role | Data entity | Application interface | Platform interface |
| **Physical Structure** | Organisation unit<br>Actor | Data store | Business application | Platform application |

# AM classification by Architecture Domain and Aspect

Avancier



# AM classification by Architecture Domain and Architect Level



| Domain Level | Business | Data/Information | Applications | Infrastructure technology |
|---|---|---|---|---|
| **Enterprise level** | **Enterprise/Business**<br>Standardisation & integration of business roles & processes<br>Business function/capability hierarchy<br>Business products & services catalogue<br>Business processes and roles<br>Etc. | **Enterprise/Data**<br>Data standardisation & integration<br>Data store & data flow catalogues<br>Maps data to business functions<br>Business data model & views of it<br>Canonical data model(s)<br>Core business data entity life cycles<br>Etc. | **Enterprise/Apps**<br>Business app standardisation & integration<br>Business app portfolio/catalogue<br>Maps business apps to business functions<br>Business app life cycles and road maps<br>Etc. | **Enterprise/Platform**<br>Platform standardisation & integration<br>Platform technology portfolio/catalogue<br>Platform services portfolio/catalogue (TRM)<br>Platform technology life cycles and road maps<br>Etc. |
| **Solution level** | **Solution/Business**<br>For a required system/solution:<br>Business services<br>Business processes and roles<br>Mappings to goals & locations<br>Requirements catalogues<br>Use case diagrams and definitions<br>Outline UI (or other I/O) designs<br>Etc. | **Solution/Data**<br>For a required system/solution:<br>Maps data to processes and roles<br>Logical data models<br>CIA requirements<br>Data qualities/meta data<br>Etc. | **Solution/Apps**<br>For a required system/solution:<br>Maps use cases to processes and roles<br>Maps business apps to use cases<br>Design for NFRs<br>Coarse-grained app components<br>Coarse-grained sequence diagrams<br>Etc. | **Solution/Platform**<br>For a required system/solution:<br>Maps platform to business apps<br>Platform technology definitions<br>Client & server node definitions<br>Design for NFRs<br>Outline deployment diagrams<br>Outline network diagrams<br>Etc. |
| **Software & technical level** | **Software/Business**<br>Detailed use case definitions<br>Detailed UI designs<br>Governs UI implementation<br>Etc. | **Software/Data**<br>Detailed database design<br>Detailed message design<br>Governs database administration<br>Etc. | **Software/Apps**<br>Detailed (fine-grained) software design<br>Governs software development<br>Etc. | **Software/Platform**<br>Detailed deployment diagrams<br>Detailed network diagrams.<br>Governs platform and network configuration<br>Etc. |

Avancier

# AM classification by Division and Interest (sketch only)

| Division / Concern | Business architecture | Software architecture | Technology architecture | Managed Operations |
|---|---|---|---|---|
| **Aims** | Business goals, constraints & standards | App. requirements constraints & standards | SLRs, constraints & standards | SLAs |
| **Stakeholders** | Customers, suppliers, employees etc. | Application users, maintainers etc. | IT users, operators etc. | IT users, suppliers etc. |
| **Processes** | Business processes and use cases | Development processes, patterns & standards | IT architecture definition | ITIL processes |
| **Organisations** | Business departments | SI department | IT architecture dep't | IT services management |
| **Locations** | Customer, supplier & employee locations | Application user & developer locations | Client and server device locations | Client and server device locations |
| **Data** | Products, services & information of all kinds | Data flow and data store formats | System management data, thresholds, etc. | Change requests, incident logs etc. |
| **Components** | Organisation units and roles/actors | Applications and components | Deployable artefacts | Deployed artefacts |
| **Technologies** | Business environment & equipment | Development & test environments | Production & DR environments | Server and network management tools |
| **Plans** | Business change estimates & plans | Application /Data change estimates & plans | Platform change estimates & plans | IT change management |

# Reference models

- [a pattern] an abstract structure or classification used to create more specific models.
- It can be a structure of components, processes or data elements.
- It is sometimes applicable to a particular industry or business domain.
- It can act as a design pattern

- FEA - US federal government
- APQC - generic business
- BIAN - banking
- SCOR - supply-chain businesses
- ProAct - retailers

- TMF for telecoms
  - eTOM – Business Architecture
  - SID – Data Architecture
  - TAM – Applications Architecture

- Industry canonical data models

- Open Group
  - TOGAF's III-RM
  - TOGAF's TRM
  - IT4IT for ITSM

# How to build your own two-dimensional EA framework?



- Pick 2 of the 5 scales below, and draw a matrix
- Then assign roles, activities and deliverables to the cells

| Focus | Time | Abstraction by… | | |
|---|---|---|---|---|
| **Domain** | **State** | **Composition** | **Idealisation** | **Generalisation** |
| **Business** | **Change request** | **Coarse-grained** | **Ideal** | **Generic** |
| Business | Baseline | Enterprise | Conceptual | Foundation |
| Data | Transition 1 | Segment | Logical | Common System |
| Applications | Transition n | Solution | Physical | Industry |
| Technologies | Target | Detailed Design | Deployed Solutions | Organisation |
| **Technology** | **Change delivery** | **Fine-grained** | **Real** | **Specific** |