Introduction to Enterprise Architecture

Purpose

- This introduction to Enterprise Architecture provides the context for enterprise architecture and information about effective architecture practice.

The Foundations Of An Architectural View

An **enterprise** is any collection of organisations that has a common set of goals. For example, an enterprise could be a government agency, a whole corporation, a division of a corporation, a single department, or a chain of geographically distant organizations linked together by common ownership.

An extended enterprise frequently includes partners, suppliers, and customers. If the goal is to integrate an extended enterprise, then the enterprise comprises the partners, suppliers, and customers, as well as internal business units.

Enterprise Architecture is:

1. *A conceptual structure used to develop, implement and sustain an enterprise architecture.*
2. *The organising logic for business processes and IT infrastructure reflecting the integration and standardisation requirements of an organisation's operating model.*
3. *A blueprint that defines the structure and operation of an organisation. The intent of an enterprise architecture is to determine how an organisation can most effectively achieve its current and future objectives.*

An **architecture is a formal description** that provides the basis for **arranging a system at a component level to guide its implementation,** such that the things within the system and the other systems that it interacts with **operate in an effective manner**. **An architect's role** is therefore to manage the structure and change in an enterprise to optimise effective operation.

The challenge is how to achieve this. This is further complicated by the issue of what constitutes an effective manner and who decides what effective means. Architects therefore have to deal with structure, components, opinions, politics and power in order to steer an enterprise towards an effective (usually evolving) architecture.

An enterprise architect works with the overall requirements / constraints of the enterprise to drive and govern the development of the (usually evolving) enterprise architecture; providing a balance of the general/global and specific/local outcomes required by that enterprise (at the relevant strategic, segment and capability levels - *TOGAF terms*).

A solution architect works with a set of specific requirements/constraints to deliver specific solutions; providing a balance between the needs of those specific requirements/constraints with the agreed wider enterprise architecture (at the relevant strategic, segment and capability levels).

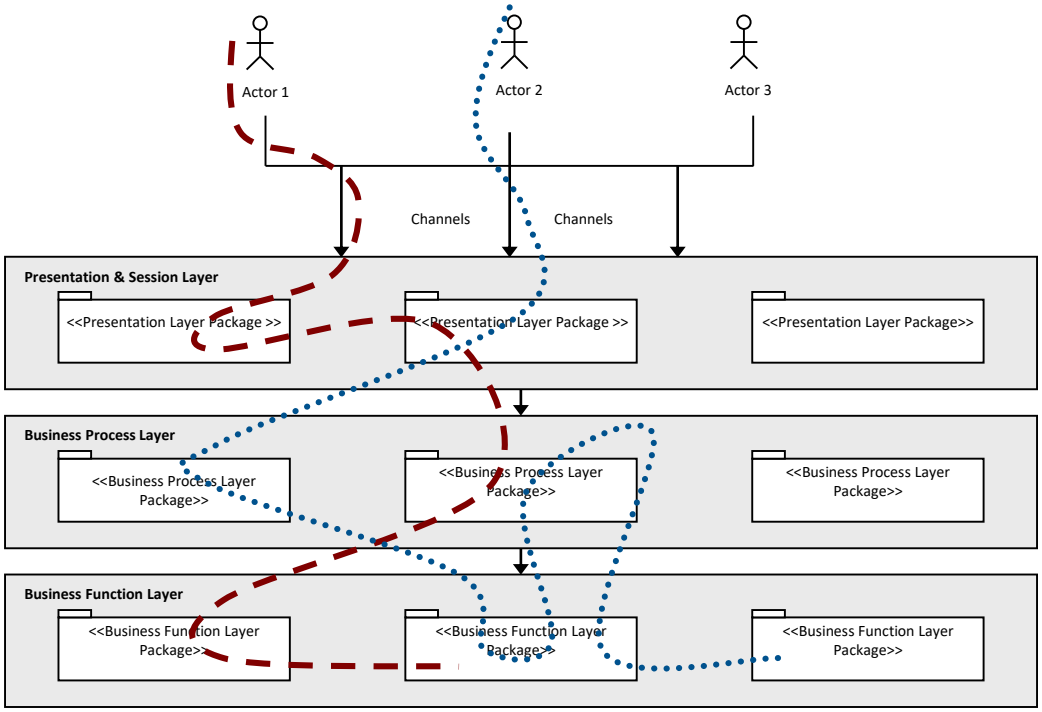
The Foundational Problem Of The Architect's Role

Managing Global vs Local Optimisation

Any solution (at the strategic, segment or capability level) will have to balance its specific impact for a particular actor and on a specific component with the wider impact on all of the other actors and components within the enterprise.

It is almost never the case that the optimum outcome for a one local viewpoint is the same for other local viewpoints or the global viewpoint.

One of an architect's unique responsibilities is to bring out this conflict, explain the issues, and work with others in the business the shape the delivered solution to do the best combination of greatest benefits and least harms.



Example: Sales in a regulated environment such as the drug industry.

Actor 1 is in sales and wishes to change a component, removing what they see as redundant steps and checks to speed up the sales process and increase yearly revenue.
Actor 2 is in compliance and wishes to add new steps and checks to ensure compliance to new traceability regulations, required to operate in the market without penalties.

Each individual set of requirements wants to make different changes to business process, applications and data. The enterprise and solution architects' role is to recognise the conflict and help to broker a solution that achieves the right balance. Note that there may be many cases where political power within the enterprise drives the chosen solution rather than optimal overall outcomes.

Architecture / Architects vs Design / Designers

The difference between the practice and roles of architecture/architects and design/designers is not always clear. In reality there are many overlaps and many people perform both or some of both roles.

Architecture without design (specific implementation and practice) is slide ware, while design without architecture (context and scope) delivers cul-de-sacs and closed off local optimisations.

Architecture and design operates at all three levels of enterprise architecture - strategic, segment and capability

Architects

Architects focus more on the abstract and logical; strategy, purpose and structure.

Designers

Designers focus more on the concrete and physical; implementation and practice.

So who is Enterprise Architecture for:

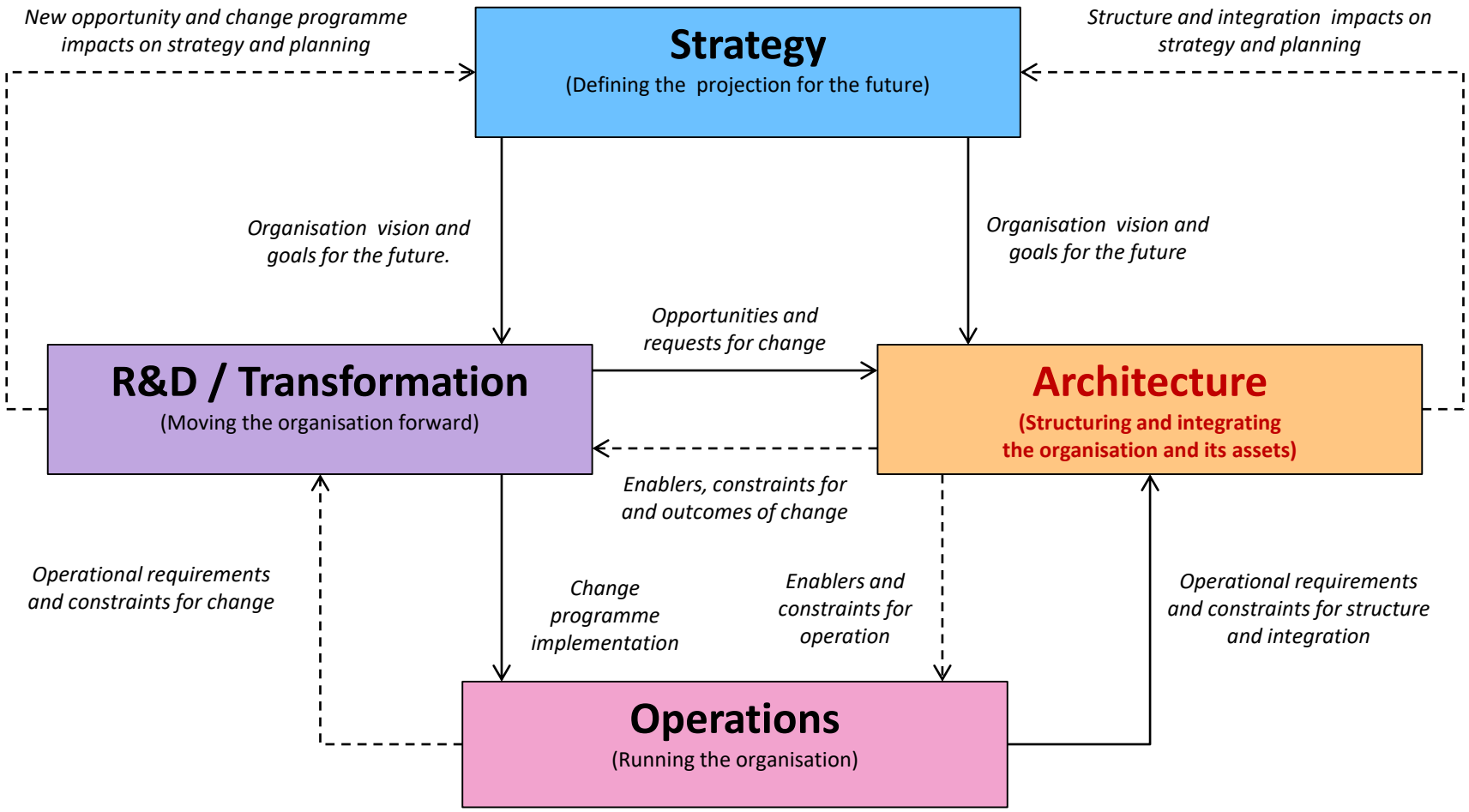
Enterprise, Business and IT Architects at all levels who construct and govern architecture building blocks (ABBs) to enable the creation of effective solution building blocks.

Business and IT Designers at all levels who need to design solution building blocks (SBBs) and must work within defined architectures.

Programme and project managers who lead projects that develop architecture and solution building blocks.

Positioning Architecture - Four Main Views Of An Organisation

Organisations are driven from four main views, that reflect the goals, change, structure, actions of and outcomes for that organisation.



Why Enterprise Architecture Is Important

All enterprises have an architecture, which is **the set of and arrangement of its resources and the manner in which they work together**.

An enterprise can manage that architecture informally or it can document it and formally manage it.

Small and relatively simple organisations can often afford to ignore the architecture and manage in a gut feel, holistic fashion.

As organisations become larger and more complex the interactions, dependencies and connections between their internal and external resources create many different effects and need to managed at a number of levels and from a number of different perspectives.

The ability of an organisation to structure, control, manage and change their resources depends on:

- The information that is held about the resources
- The accuracy and completeness of that information
- The processes that ensures the on-going management and control of that information
- The organisation's ability to use that information effectively to make good decisions about change in support of the business goals that are supported by these properties

Enterprise architecture provides the basis for defining and then managing an organisation's resources, their interactions and their outcomes. An enterprise architecture is a model of an organisation's business assets that reflects:

- The current state of those assets and their interaction
- The target state of those assets and their interaction
- The transitional states as transformation and change programmes and projects are implemented

Having effective documentation of these perspectives enables an organisation to achieve enterprise alignment and integration in order to manage change, optimise cost and value, improve quality and satisfaction, and reduce "time-to-market".

Enterprise Architecture Helps To Manage Complexity

Achieving agreement across an organisation about these connections, interdependencies and constraints is difficult.

Most business managers are primarily focused on running their existing business processes.

- They are usually measured on immediate exploitation of those business process; rather than the overall management of the processes themselves over time.
- This makes it difficult to enable the organisation to grow, change and improve in line with the strategic direction set by the executive.

Enterprise architecture provides a basis for understanding all of the resources in an organisation and considering both the immediate tactical and future strategic implications.

Finding time to define and mange the enterprise architecture itself and its implications is not easy. Operational managers tend not to have the time to for this and only become involved with it when they:

- Have a failure or a problem
- Wish to do something they have not done before
- Have a failure of problem in using their existing resources to achieve their goals
- Have to deal with negotiation, boundary, transition, alignment issues

It is usually only when these problems arise that the underlying architecture of an organisation becomes visible.

It is similar to the attention we apply to plumbing or electricity system, we assume it all works until it fails.

The benefits of good architectural definition and control can thus be difficult to justify to the business over the short term.

However, dealing with complex operational problems and achieving strategic goals requires a good understanding of the architecture of an organisation and how to evolve it in line with a careful balancing to tactical and strategic goals and outcomes.

Enterprise Architecture Address Structure and Interaction

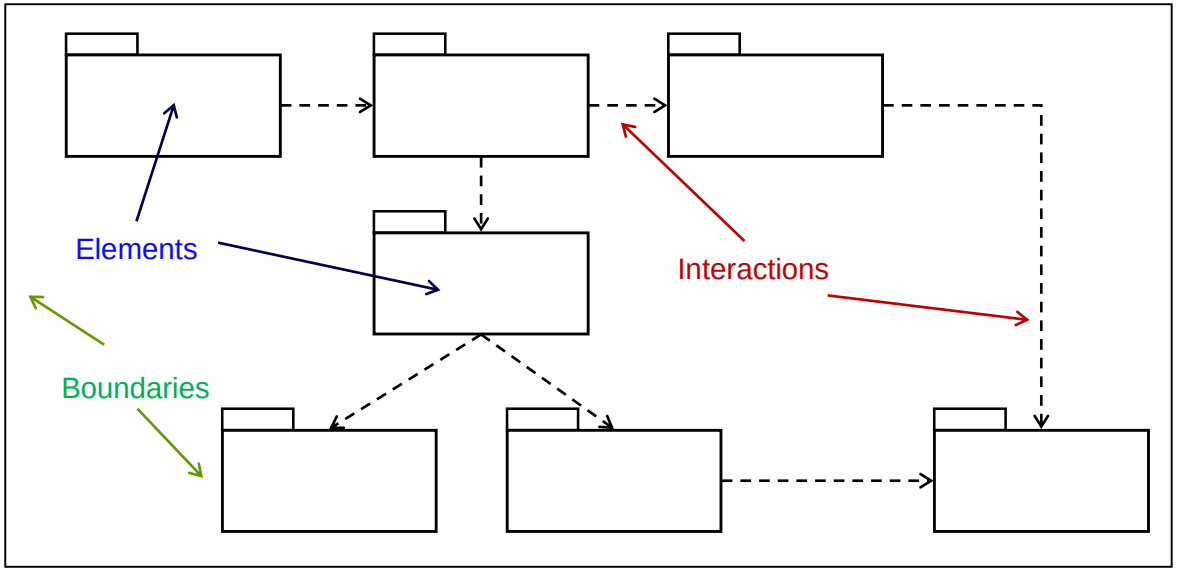
Enterprise architecture identifies the main structures within an organisation and how they interact.

In a “technical sense” it is about the elements (resources) within an organisation and their:

- Properties
- Arrangement
- Boundaries
- Interactions and relationships

Different parts of an organisation often have different and conflicting goals and practices that need to operate together in a collective ecosystem.

The purpose of enterprise architecture is to optimise across the enterprise the often fragmented legacy of processes and resources (both manual and automated) into an integrated environment that is responsive to change and supportive of the deliver y of the business strategy.



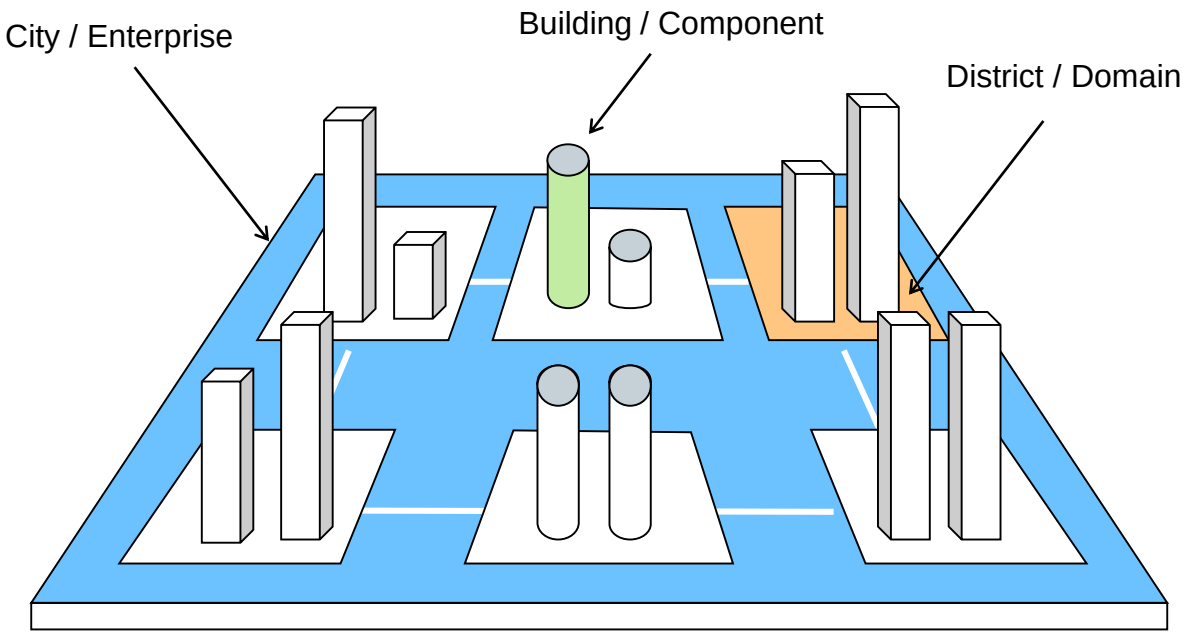
Enterprise architecture provides the blueprint, plan or model and the management practices to more actively and effectively manage the complexities within an organisation by facilitating the:

- Definition of the resources within a business
- Definition of the boundaries and relationships between these resources
- Negotiation and management across these boundaries
- Management of the shared processes and systems across these boundaries
- Establishment of trust across these boundaries
- Establishment of responsibilities for resources within the business
- Governance for those resources from the perspective of the complete business ecosystem.

Different Contexts For Architecture

City planning is an often used analogy for enterprise architecture. In simple terms a city can be considered to have at least three organising contexts.

- City (Enterprise)
  The structures and integration needed across the whole city (sewers, roads etc.).
- District (Domain/Segment)
  The structures and integration needed to manage the district and its buildings and integrate back into the city wide structure.
- Building (Component/Capability)
  The structures needed to deliver the specific properties of the building and integrate back into the district structure.



The City Planning Analogy

Different capabilities are planned and managed at the different levels .

- Roads, sewers, water pipes electricity grids tend to be planned across the city
- Types of activity, (business, leisure, housing) are often “zoned” into specific districts.
- Individual buildings are then designed within these contexts and to meet their specific requirements.

City planning sets the context and the basis for integration for district and building planning.

The Benefits Of Enterprise Architecture

To manage an enterprise we therefore need to create:

- An understanding/model of the resources available to the enterprise
- A set of activities that utilise that understanding/model to make better decisions about those resources and how thy are used

This provides the basis for managing change to an enterprise and delivering business benefit:

- Specifically via a business case for each change
- More generally through wider rationalisation, reuse, cost reduction, simplification and specialisation of the business resources.

Enterprise architecture provides the models and processes for this.

The deliverables from an effective enterprise architecture are:

- A clear, well documented definition of the current state of an organisation's resources.
- Well defined current, transition and target architectures that provide the context for change and speed decision making and impact assessments.
- Reusable and standardised components within the architecture that drive cost reduction and quality improvements.
- An understanding of the relationship between planned change and current operation that minimises implementation errors for each change project.
- A clear architectural view and high level design for each specific change that improves the ability to assess costs, risk and impact before the change is budgeted and approved.

The benefits from an effective enterprise architecture are:

Better knowledge and management of business assets and their interaction, reducing the cost of change and enabling that change to be implemented:

- Individually with less risk and more speed
- Collectively with improved alignment:
  - Between each change initiative
  - To business strategy and goals

Key Concepts In TOGAF

Who Are The Open Group?

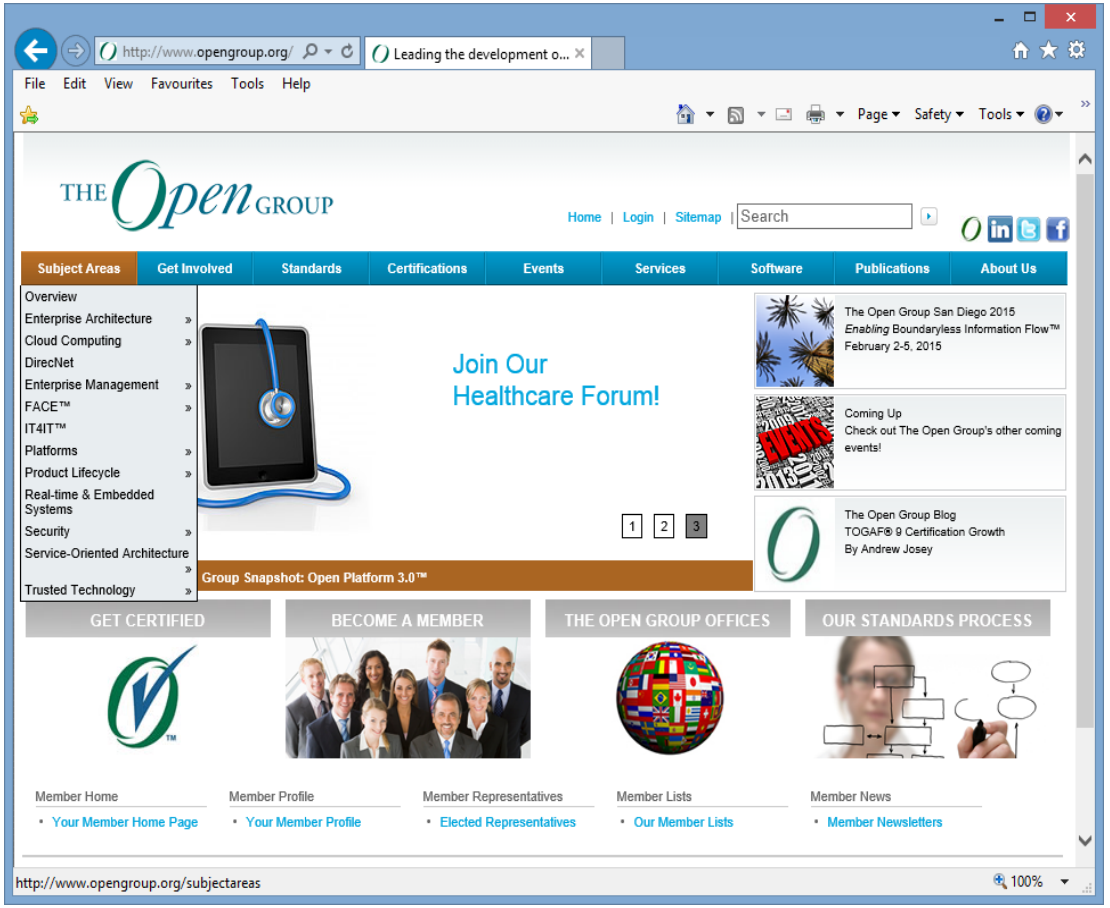
The Open Group is a consortium. It was founded in 1996 when X/open merged with the OSF.

It's initial focus was on technical standards such as UNIX/POSIX, CORBA, WAP, ODBC, LDAP.

It now focuses on open standards and the flow of information across organisations.

One of the standards is the Open Group Architecture Framework (TOGAF).

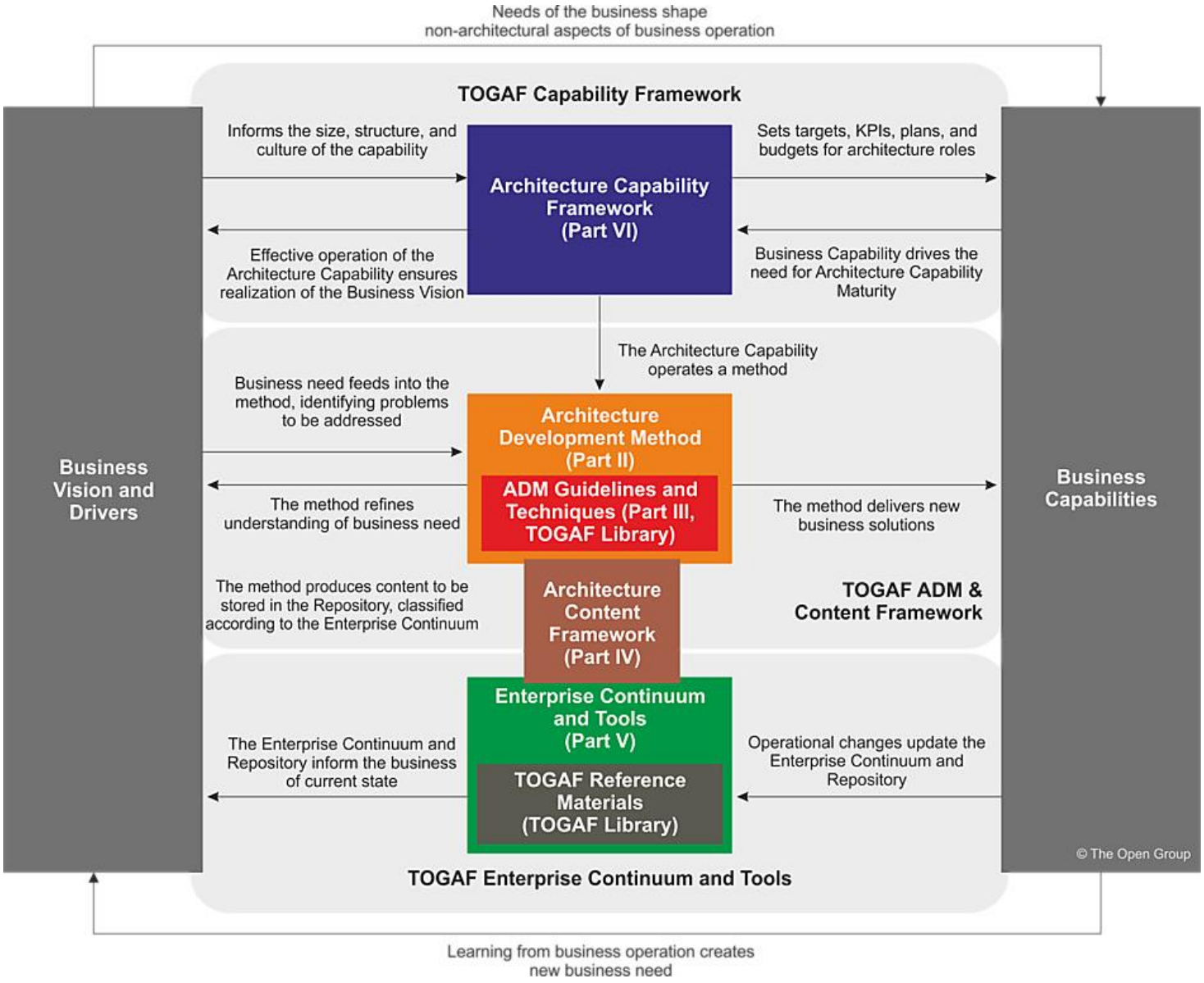
The Open Group covers a diverse group that spans all sectors of the ICT community ICT customers, systems and solutions suppliers, tool vendors, integrators and consultants, as well as academia and researchers.



There are 250 member organisations come from all over the world; 50% from North America, 25% from Europe, and 25% from Asia-Pacific. These include Capgemini, Fujitsu, Hewlett-Packard, Hitachi, IBM, PWC, US Department of Defence and NASA.

Many government departments and agencies use TOGAF particularly in Europe, and the US.

An Overview Of TOGAF



The Architecture Framework

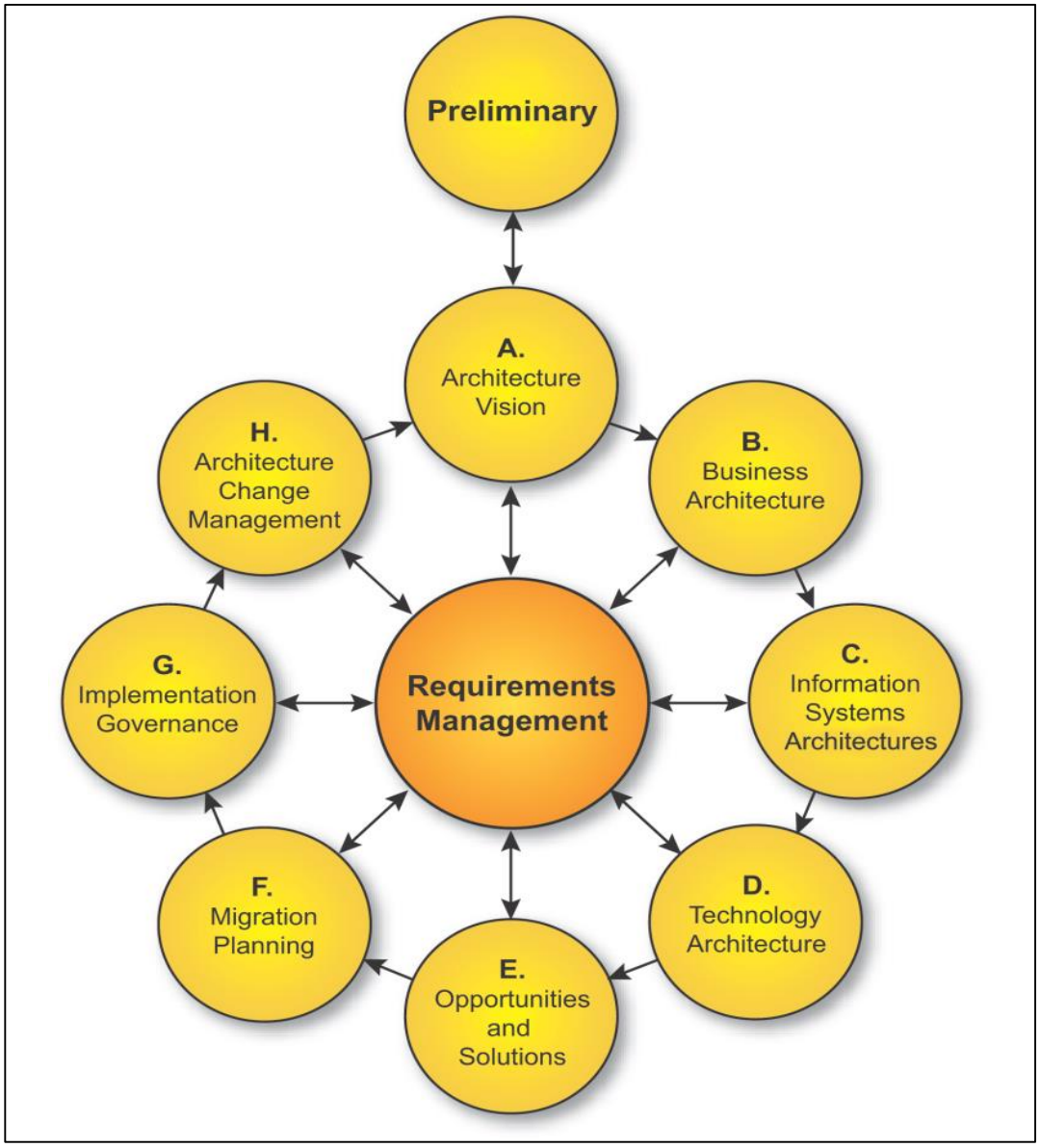
The Open Group Architecture Framework (TOGAF) is a tool for assisting in the acceptance, production, use, and maintenance of enterprise architectures

It is based on an iterative process model supported by best practices and a re-usable set of existing architectural assets.

It is developed and maintained by The Open Group Architecture Forum.

The first version of TOGAF, developed in 1995, was based on the US Department of Defence Technical Architecture Framework for Information Management (TAFIM).

It has also incorporated elements from most of the other major architecture approaches from across the industry (e.g. Zachman, DODAF, MODAF, FEAF) into an overall consistent framework and development process.



The Four Major Viewpoints Of The Architecture Framework

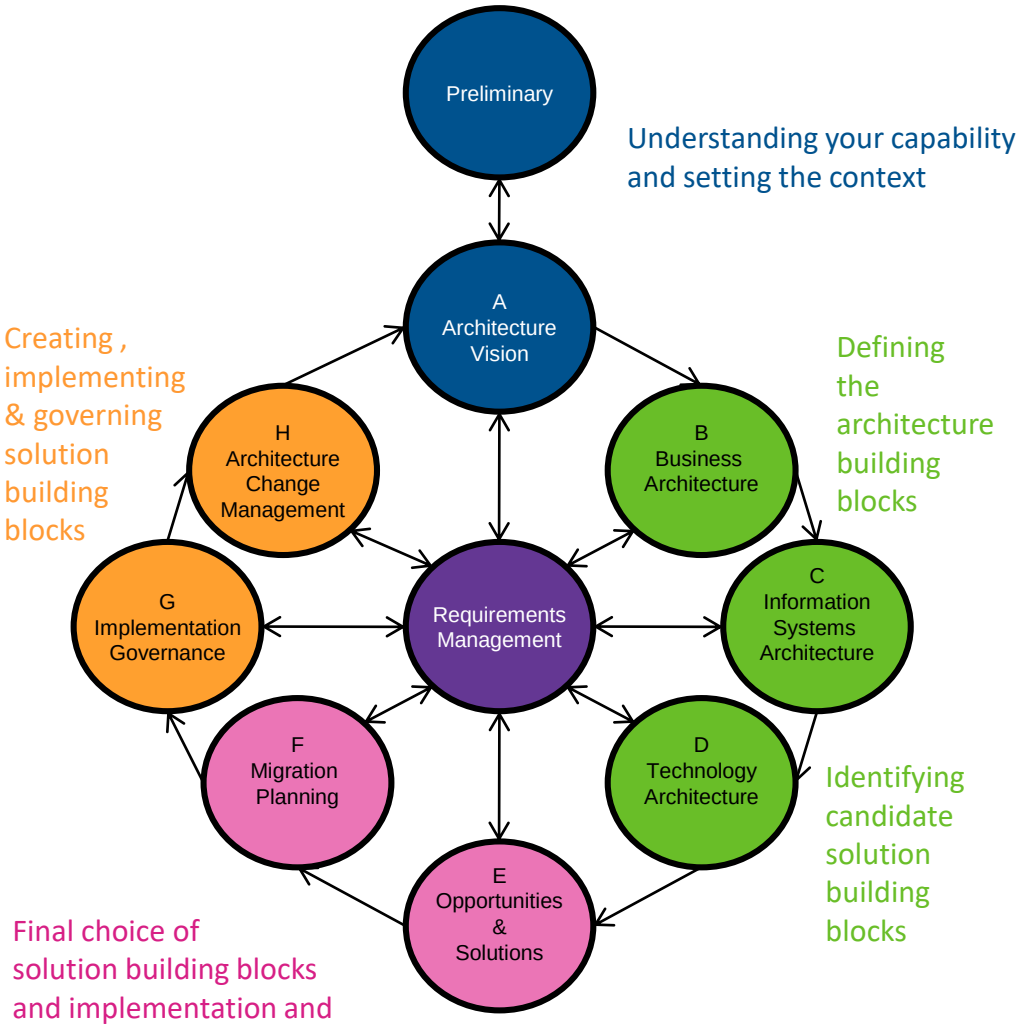
The latest version, TOGAF 9, was introduced in 2009. It can be used for developing a broad range of different enterprise architectures. TOGAF complements, and can be used in conjunction with, other frameworks that are more focused on specific deliverables for particular vertical sectors such as Government and Telecommunications. TOGAF is

- A framework, not an architecture
- A generic framework for developing architectures to meet different business needs
- Not a “one-size-fits-all” architecture
- Focuses on business IT alignment
- Based in best practices
- Widely adopted in the market
- Tailorable to meet an organisation and industry needs

It is broken down into phases that have a continuous flow into each other as an organisation evolves its capability and the associated architecture views.

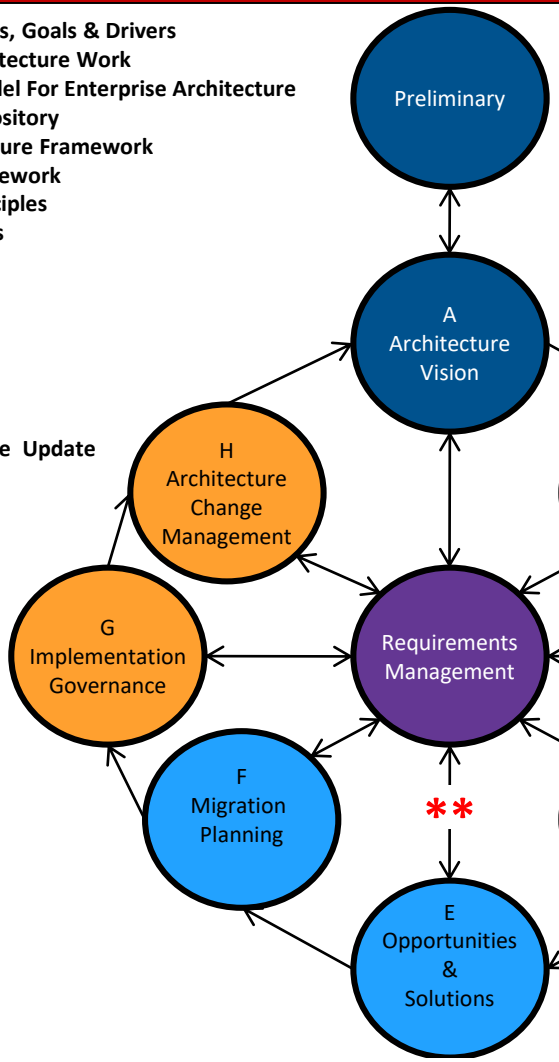
The specific architectural elements within TOGAF are:

- Business architecture
- Information systems architecture
- Technology architecture



Document Deliverables Around the Lifecycle (Outputs)

Each phase can generate a set of deliverables to help manage the evolution of the enterprise architecture and each specific change introduced into the organisation.



Business Principles, Goals & Drivers
Request For Architecture Work
Organisation Model For Enterprise Architecture
Architecture Repository
Tailored Architecture Framework
Governance Framework
Architecture Principles
Architecture Tools

Approved Statement of Architecture Work
Communications Plan
Tailored Architecture Framework
Architecture Vision
Architecture Principles
Capability Assessment
Vision Architecture Definition Document
Vision Architecture Requirements Specification
Vision Candidate Roadmap

Managed Architecture Update

Approved Statement of Architecture Work
Draft Architecture Definition Document (Business)
Draft Architecture Requirements Spec. (Business)
Business Architecture Principles
Candidate Roadmap

Signed Architecture Contract
Solution Building Block
Compliance Assessment
Architecture Compliant Solution
Governance Framework
Change Request

Approved Statement of Architecture Work
Draft Architecture Definition Document (IS)
Draft Architecture Requirements Spec. (IS)
IS Architecture Principles
Candidate Roadmap

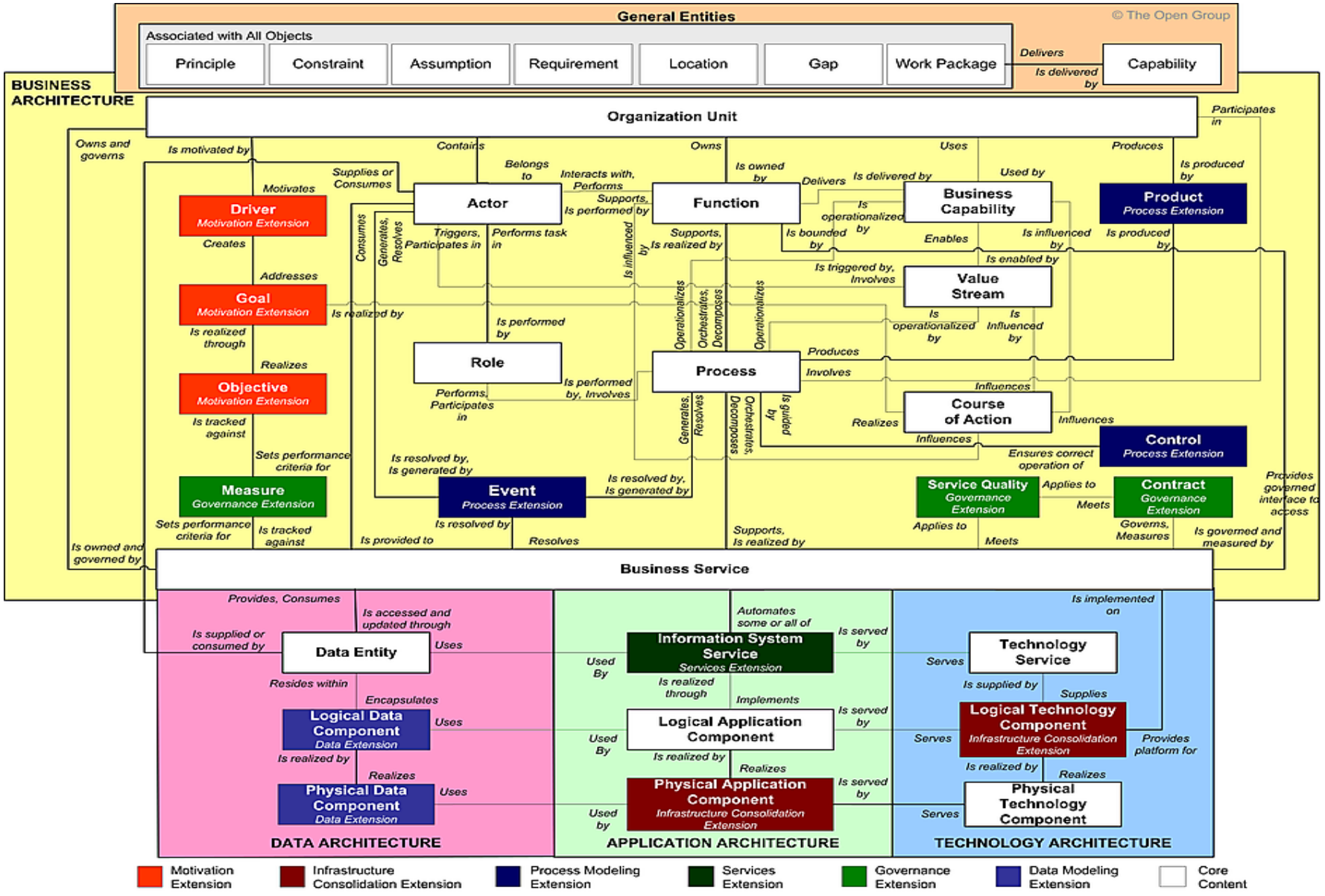
Request For Architecture Work
Final Architecture Definition Document
Final Architecture Requirements Spec.
Final Architecture Roadmap
Final Transition Architecture
Draft Architecture Contract
Implementation & Migration Plan
Implementation & Migration Strategy
Implementation Governance Model
Finalised Architecture Building Block
Change Request

Approved Statement of Architecture Work
Draft Architecture Definition Document (Technology)
Draft Architecture Requirements Spec. (Technology)
Technology Architecture Principles
Candidate Roadmap

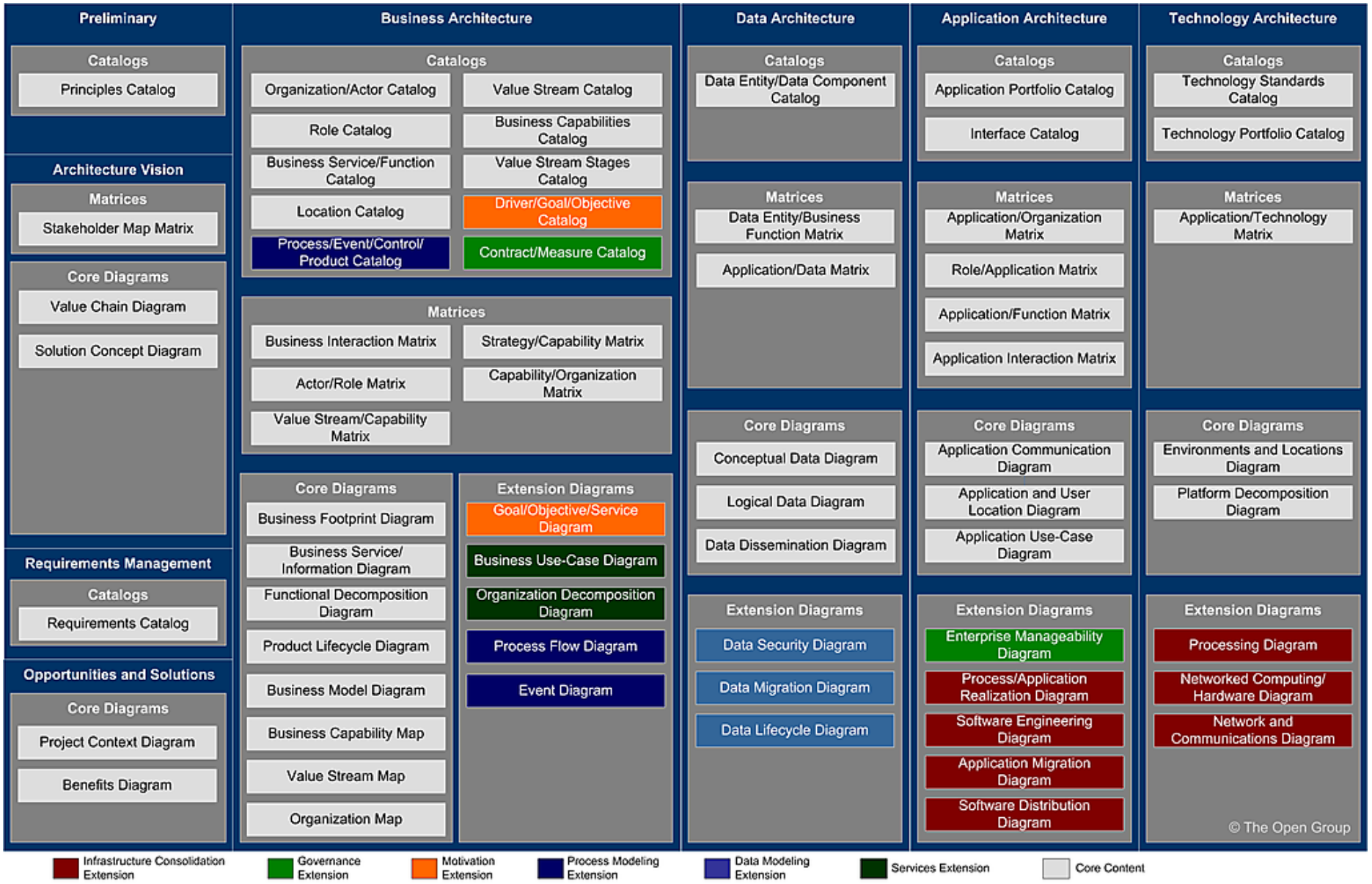
Architecture Requirement
Requirements Impact Assessment
**

Approved Statement of Architecture Work
Draft Architecture Definition Document
Draft Architecture Requirements Spec.
Consolidated Architecture Roadmap ( ) **Transition Architecture**
(Initial) **Implementation & Migration Plan**
(Initial) **Implementation & Migration Strategy**

The Content Framework Metamodel



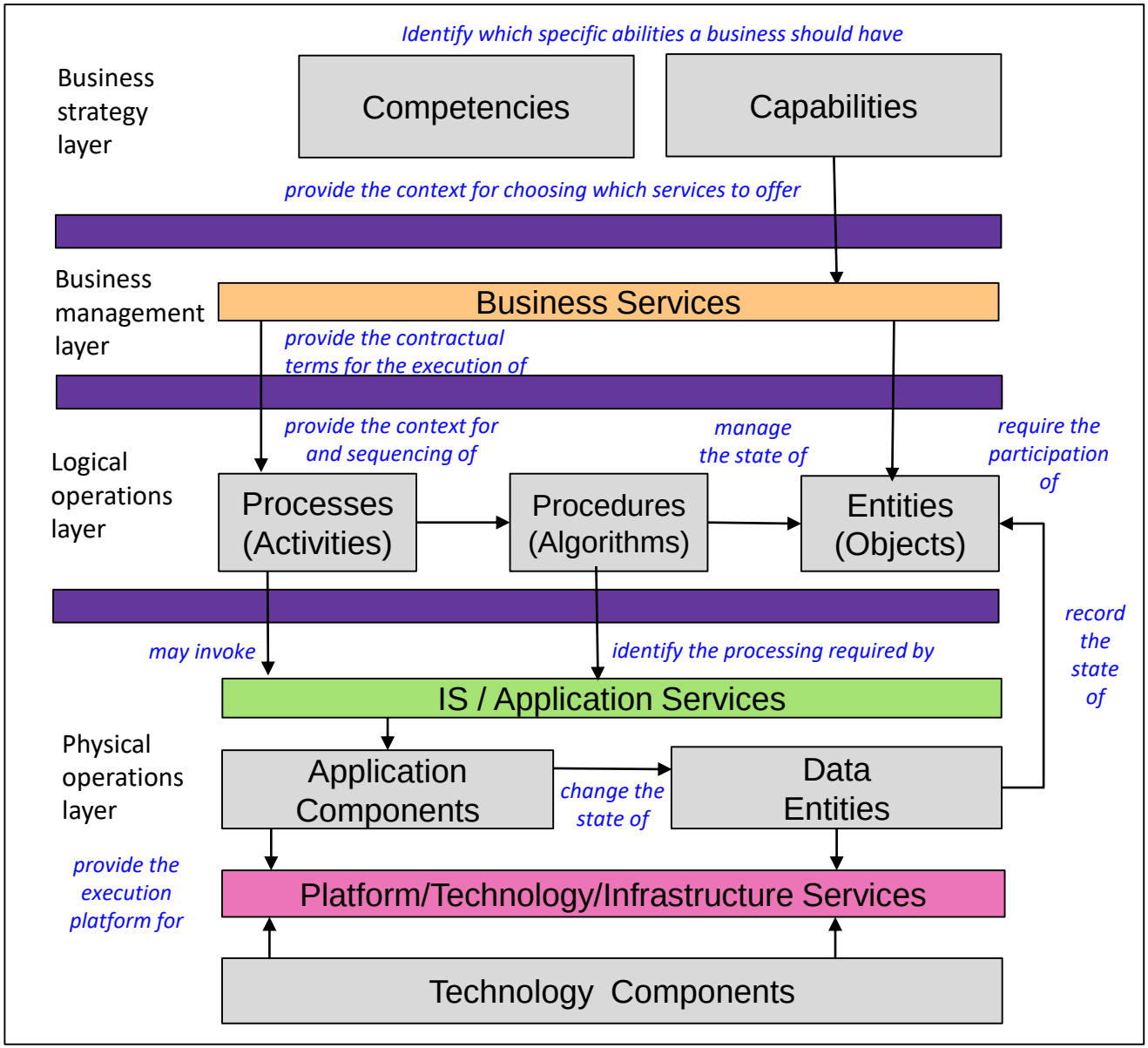
Example Specific Architecture Artefacts In The Content Framework L1 / L2



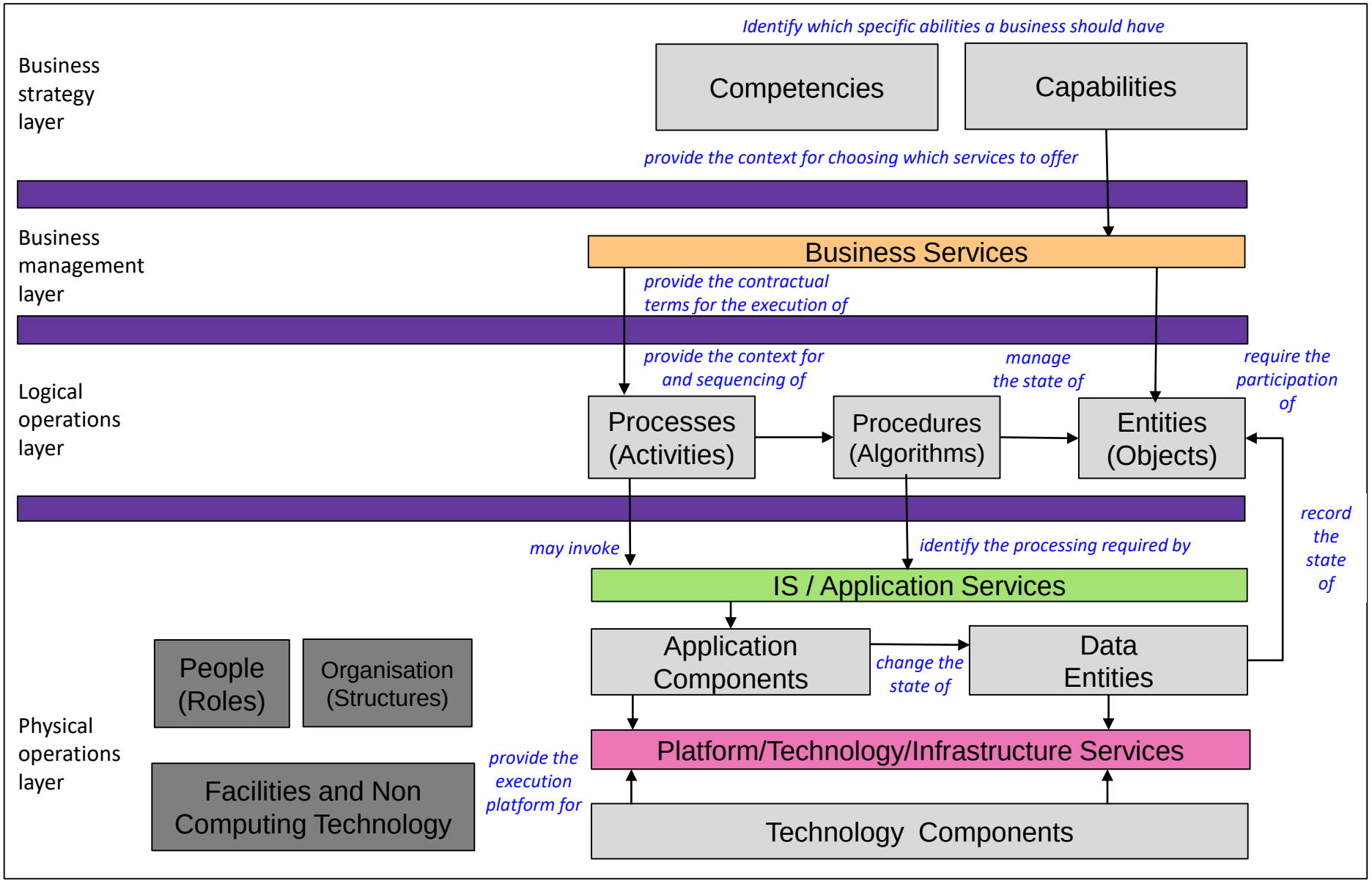
Making Sense Of The Capabilities and Layering

L1 / L2

- The TOGAF content framework differentiates between the processes of a business and the services of a business.
  - Business services are specific processes that have explicit, defined boundaries that are explicitly governed.
  - Services are distinguished from processes through the explicit definition of a service contract that defines a post condition and it performance attributes.
- The granularity of business services should be determined according to the business drivers, goals, objectives, and measures for this area of the business. Finer-grained services permit closer management and measurement (and can be combined to create coarser-grained services), but require greater effort to govern.



Completing The Business Architecture View (POPIT)

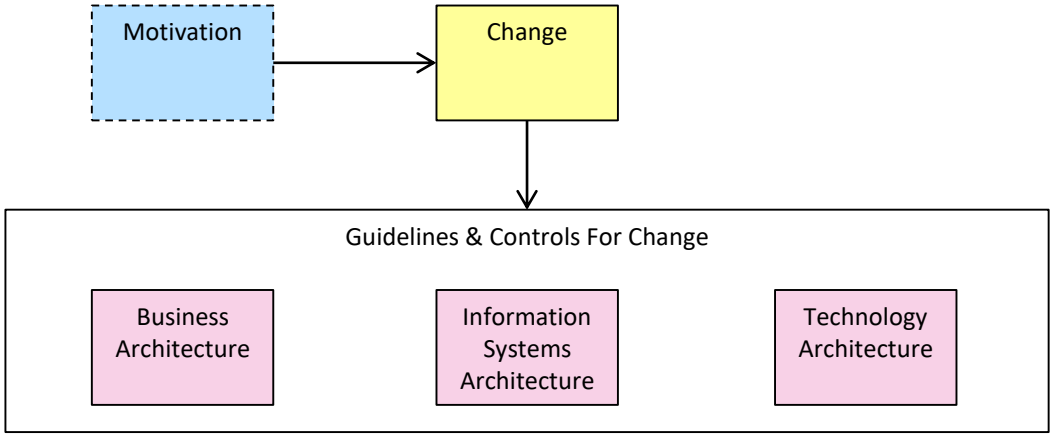


Good Architecture Practice

Choosing An Approach For Enterprise Architecture

An enterprise architecture needs to be organised to support effective definition and management. As an organisation:

- Changes its motivation (vision, strategy and goals)
- A series of changes may be required to the resources within the enterprise.
- A set of guidelines to manage that change in a coordinated and consistent fashion is required



An enterprise architecture provides a structure within which the resources subject to change can be identified, viewed and modified to achieve the best balance of longer term strategic and immediate tactical outcomes. The most widely used approach for managing an enterprise architecture is **The Open Group Architecture Framework (TOGAF).**

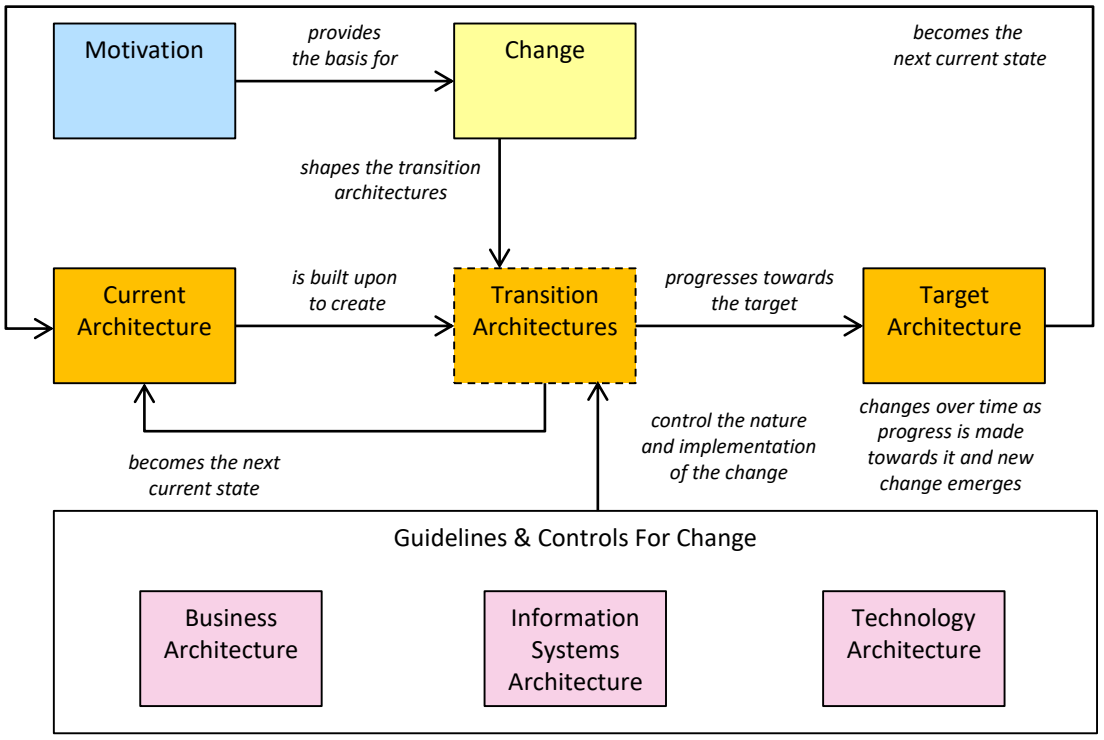
TOGAF's initial organising concept is the grouping of business resources into three specific sets.

- **Business architecture** - organisation structure, operating model, party/people, processes, information
- **Information systems architecture** - systems, applications and data
- **Technology architecture** - computing & network devices, software products, other physical and technologies

The Evolution Of An Enterprise Architecture

An enterprise architecture is not static but changes over time. This can be viewed as a series of state definitions representing:

- A current/baseline architecture (the current state)
- Transition architectures (steps on the path towards the target architecture)
- A target architecture (the future desired state)



The concept of a current/baseline, transition and target state applies to all of the components within an enterprise architecture. Each organisation decides which components to formally manage and may create a series of change roadmaps. The most commonly found roadmaps tend to be for:

- Technology and application evolution (sometimes combined as platforms)
- Business, software and physical service evolution
- Process evolution

Identifying Solution Requirements And Specification Properties

Architecture and design requires an understanding of the desired properties of the solution; as expressed in the requirements and then in the specification (that it is agreed can be developed to best [often not completely] meet those requirements).

The table to the right shows the different types of requirements that may be either explicitly or implicitly expressed.

It is rare that all of these are specified in the requirements but all of these will have a performance level in the final solution.

Architecture and design activity must ensure that the performance levels for these properties are understood and the agreed performance levels are delivered by all of the solution components working together.

Business Concerns	**Sellability**	**Operability**	**Sustainability**	**Functionality**	
	Marketability	Runability	Carbon Footprint	Purpose	
	Profitability (Cost/Price)	Repairability	Power Requirements	Feature	
	Longevity	Instrument-ability	Space Footprint	Action/Task	
	Deliverability (Time - Market)	Reliability	Water Requirements	Rule	
Agility Concerns	**Capability Maturity**	**Testability**	**Adaptability**	**Conceptual Integrity**	
	Technology	Clarity Of Requirements	Extendibility	Strategic Alignment	
	Skills	Clarity Of Design	Reusability	Pattern	
	Experience	Simplicity	Portability	Style	
	Process	Scalability	Modularity	Standardisation	
End User Concerns	**Security**	**Availability**	**Capacity**	**Usability**	**Accuracy**
	Auditability	Responsiveness	Throughput	Learnability	Atomicity
	Confidentiality	Recoverability	Storage	Likeability	Consistency
	Visibility & Detectability	Integrity	Volume	Productivity	Integrity
	Accessibility	Installability	Scalability	Dexterity	Durability

Matching The Requirements And Solution Specification Properties

The list on the left shows the properties that are initially requested.

It is often not possible to meet all of these within the same architecture / design (e.g. a requirement to deliver recoverability may conflict with one for responsiveness).

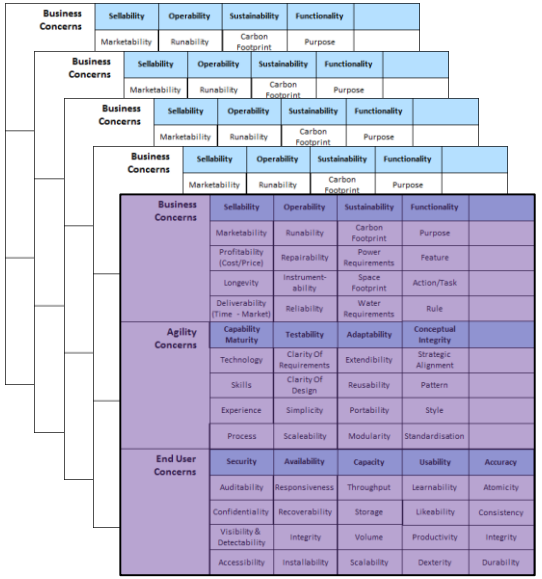
The lists to the right show properties of the different solutions that may be able to meet some or all of those requirements. There will be few cases where there is a perfect match between the properties of the requirements and the properties of the solutions.

Customer / service / product solution requirement properties that may be contradictory between themselves

Business Concerns	Sellability	Operability	Sustainability	Functionality	
	Marketability	Runability	Carbon Footprint	Purpose	
	Profitability (Cost/Price)	Repairability	Power Requirements	Feature	
	Longevity	Instrument-ability	Space Footprint	Action/Task	
	Deliverability (Time - Market)	Reliability	Water Requirements	Rule	
Agility Concerns	**Capability Maturity**	**Testability**	**Adaptability**	**Conceptual Integrity**	
	Technology	Clarity Of Requirements	Extendibility	Strategic Alignment	
	Skills	Clarity Of Design	Reusability	Pattern	
	Experience	Simplicity	Portability	Style	
	Process	Scaleability	Modularity	Standardisation	
End User Concerns	**Security**	**Availability**	**Capacity**	**Usability**	**Accuracy**
	Auditability	Responsiveness	Throughput	Learnability	Atomicity
	Confidentiality	Recoverability	Storage	Likeability	Consistency
	Visibility & Detectability	Integrity	Volume	Productivity	Integrity
	Accessibility	Installability	Scalability	Dexterity	Durability

Architecture and design is the act of balancing the requirements to reflect possible combinations and then matching them to the properties of potential solution components.

Different existing component / supplier properties that may meet some or all of the required solution properties

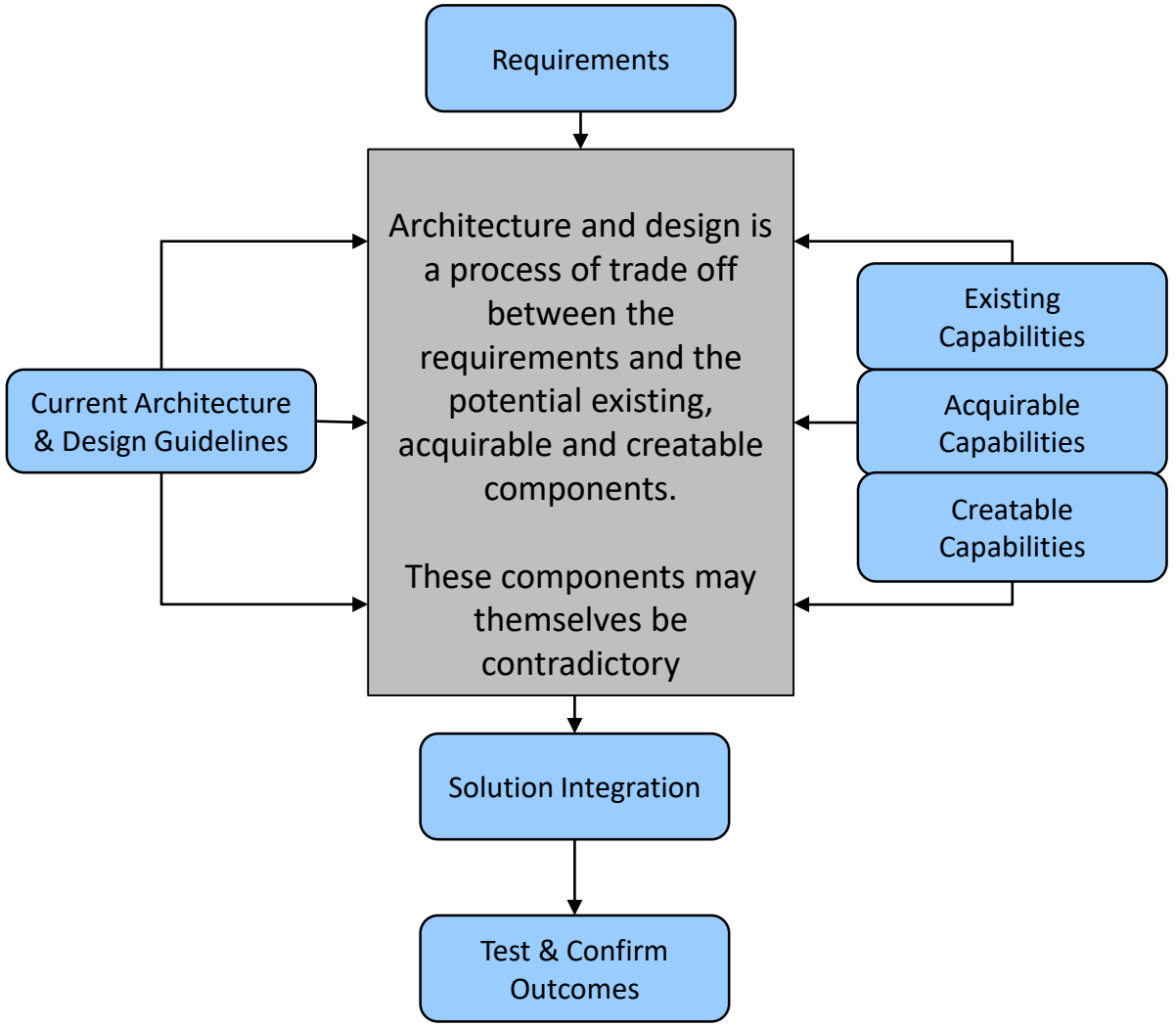


One of the possible sets of real implementable properties has to be chosen.

Architecture And Design Is A Series Of Trade Offs

As architecture and design progresses the differences and contradictions between the desired and available properties emerge.

Architecture and design is therefore often a process of trading off the performance of different requirements and the ability of different potential solutions to best meet those requirements.



Service Oriented Architecture

Architecture and design is about bringing different components together to deliver a set of outcomes.

Service Oriented Architecture (SOA) is an approach to both business and technical services based on a set of principles about how components can be defined and interact in a managed and visible manner (based on ideas developed about contracts).

Those SOA principles provide the basis for defining and implementing services that utilise integrated components to deliver specified business outcomes.

A **service**
represents the delivery of an agreed outcome, by a provider within the bounds of defined performance, quality and service levels; in response to a request by a client.

A **business service** delivers a specific outcome to a customer using the relevant resources of a business (people, organisation, process, information and technology)

In some cases the business service can be delivered by just using one or more software services, in other situations additional elements such as visiting buildings, talking to people, delivering or installing physical devices; may also be needed

A **software service** delivers a specific outcome in terms of an internal information system state change and/or a return of information; in response to an identified system request/event

The General Properties Of A Service

The "SOA Manifesto Working Group", have defined a SOA Manifesto.

Its aim is to provide clarity about the basic properties of a good service oriented architecture.

The manifesto provides set of overall general tendencies for a good service oriented architecture that value:

- Business value over technical strategy.
- Strategic goals over project-specific benefits.
- Intrinsic interoperability over custom integration.
- Shared services over specific-purpose implementations.
- Flexibility over optimization.
- Evolutionary refinement over pursuit of initial perfection.

These are underpinned by a set of basic properties that should apply to the components within a SOA:

- Reuse
- Granularity
- Modularity
- Composability
- Componentization
- Interoperability
- Standards-compliance (both common and industry-specific)
- Service identification and categorization.

The Specific Properties Of Effective Services

SOA tends to focus on the properties of software services. The properties listed show key aspects of a well formed software services.

These aspects are equally important when applied to business services (within which the software services operate).

As an architect or designer you should always be aware of these aspects when designing the integration of components for contracts, products and services. The more that each of these compositions support these aspects the greater the potential for future reuse.

- **Standardized Service Contract** – Services adhere to an agreed contract, as defined collectively by one or more service-description documents.
- **Service Loose Coupling** – Services maintain a relationship that minimizes dependencies and only requires that they maintain an awareness of each other.
- **Service Abstraction** – Beyond descriptions in the service contract, services hide logic from the outside world.
- **Service Reusability** – Logic is divided into services with the intention of promoting reuse.
- **Service Autonomy** – Services have control over the logic they encapsulate.
- **Service Granularity** – A design consideration to provide optimal scope and the right granular level of the business functionality in a service operation.
- **Service Statelessness** – Services minimize resource consumption by deferring the management of state information when necessary
- **Service Discoverability** – Services are supplemented with communicative meta data by which they can be effectively discovered and interpreted.
- **Service Composability** – Services are effective composition participants, regardless of the size and complexity of the composition.
- **Service Normalization** – Services are decomposed and/or consolidated to a level of normal form to minimize redundancy. In some cases, services are denormalised for specific purposes, such as performance optimization, access, and aggregation.
- **Service Optimization** – All else equal, high-quality services are generally preferable to low-quality ones.
- **Service Relevance** – Functionality is presented at a granularity recognized by the user as a meaningful service.
- **Service Encapsulation** – Many services are consolidated for use under the SOA. Often such services were not planned to be under SOA.
- **Service Location Transparency** – This refers to the ability of a service consumer to invoke a service regardless of its actual location in the network

Heuristics For Good Architecture And Design

A heuristic is :

A distillation of experience about a way of working that has been found to be of value.

By following good heuristics you improve our overall architecture and design practice.

The core heuristics for good architecture and design are:

- Understand your context
- Produces satisficing (good enough) designs
- Manage dependent outcomes
- Prioritise your activities and deliverables
- Actively manage risks
- Simplify, simplify (but don't over simplify)
- Create modules
- Build solutions from available modules
- Focus e2e solution design on the interfaces
- Use agile / iterative / evolutionary development lifecycles
- Deliver frequent deployments based on short sprints / cycles
- Develop stable intermediates
- Design for testability and test often
- Do not optimize too early

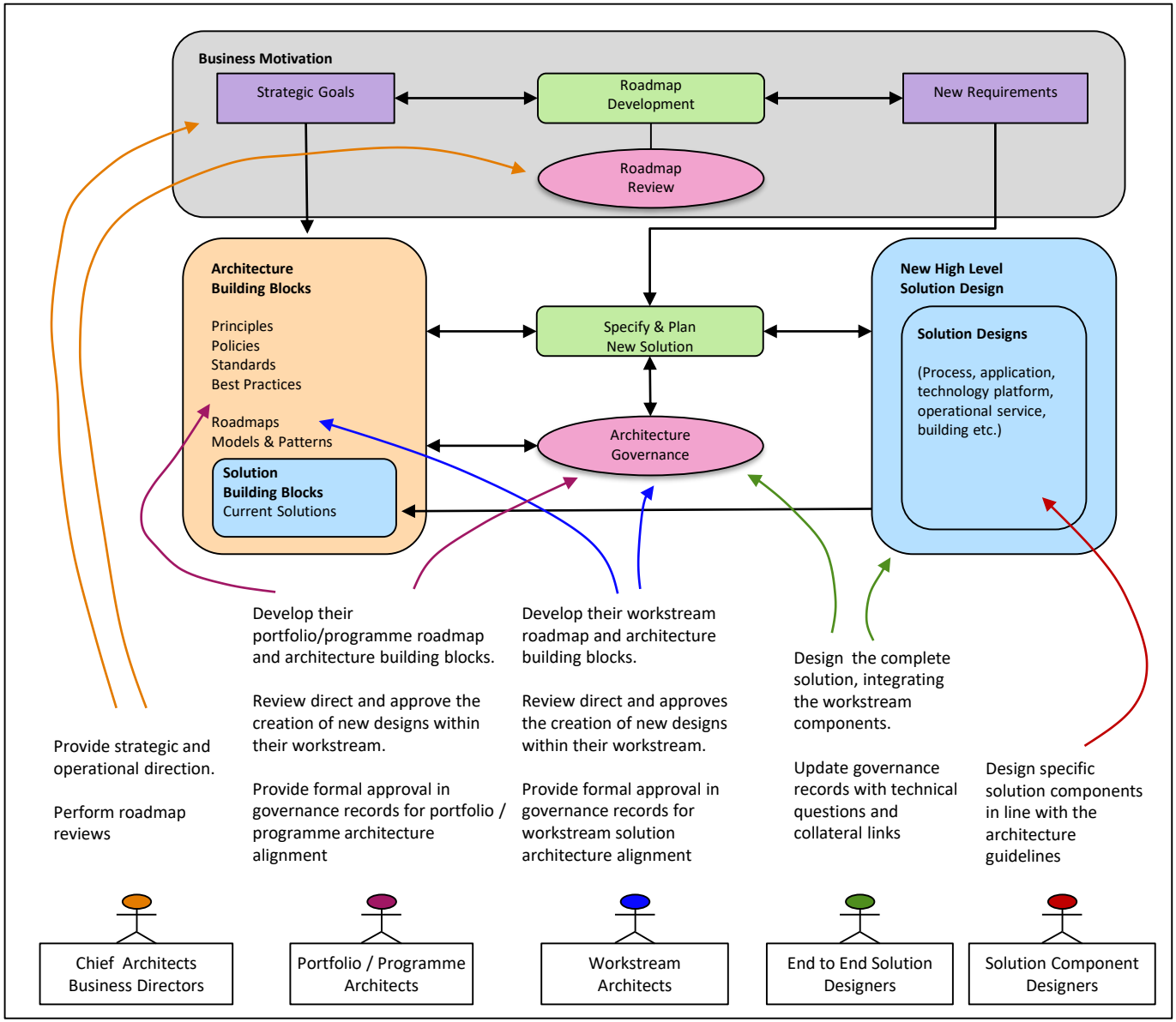
Example Process For Delivering Architectural Alignment

Delivering change that is aligned to an enterprise architecture is a dynamic process supported by a number of roles.

Example roles and their responsibilities are shown.

Architecture governance provides the basis for new and changing architectures and designs to reflect the right balance of architectural conformance and specific requirements for an individual solution.

(Note the term “workstream” in the graphic represents each set of different elements that make up a solution (e.g. process, application platform, operational service, buildings etc.).

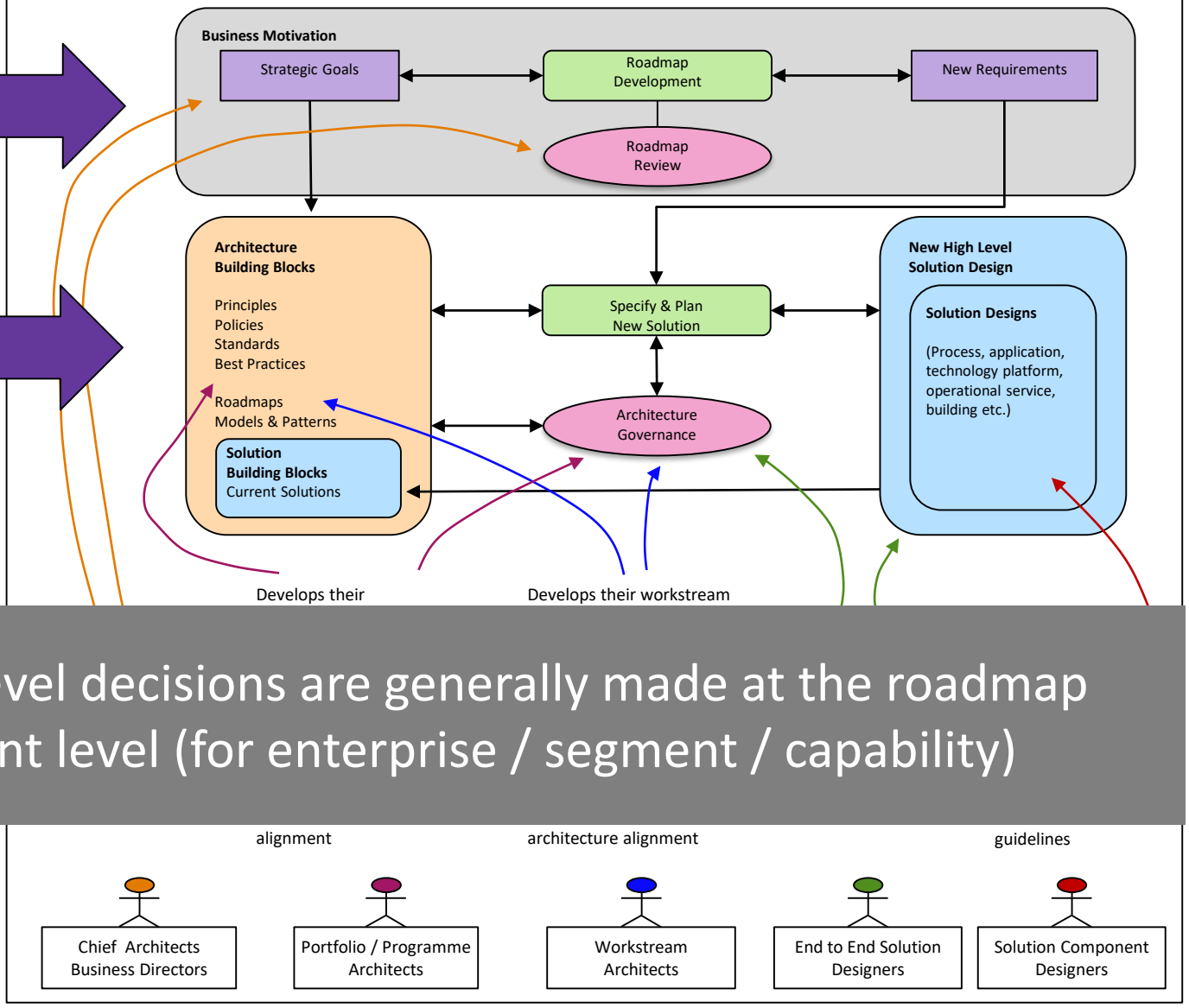


Example Process For Delivering Architectural Alignment - AL

Delivering change that is aligned to an enterprise architecture is a dynamic process supported by a nu of roles.

Example roles and their responsibilities are show

Architecture governance provides the basis for ne changing architectures a designs to reflect the righ balance of architectural conformance and specifi requirements for an indiv solution.



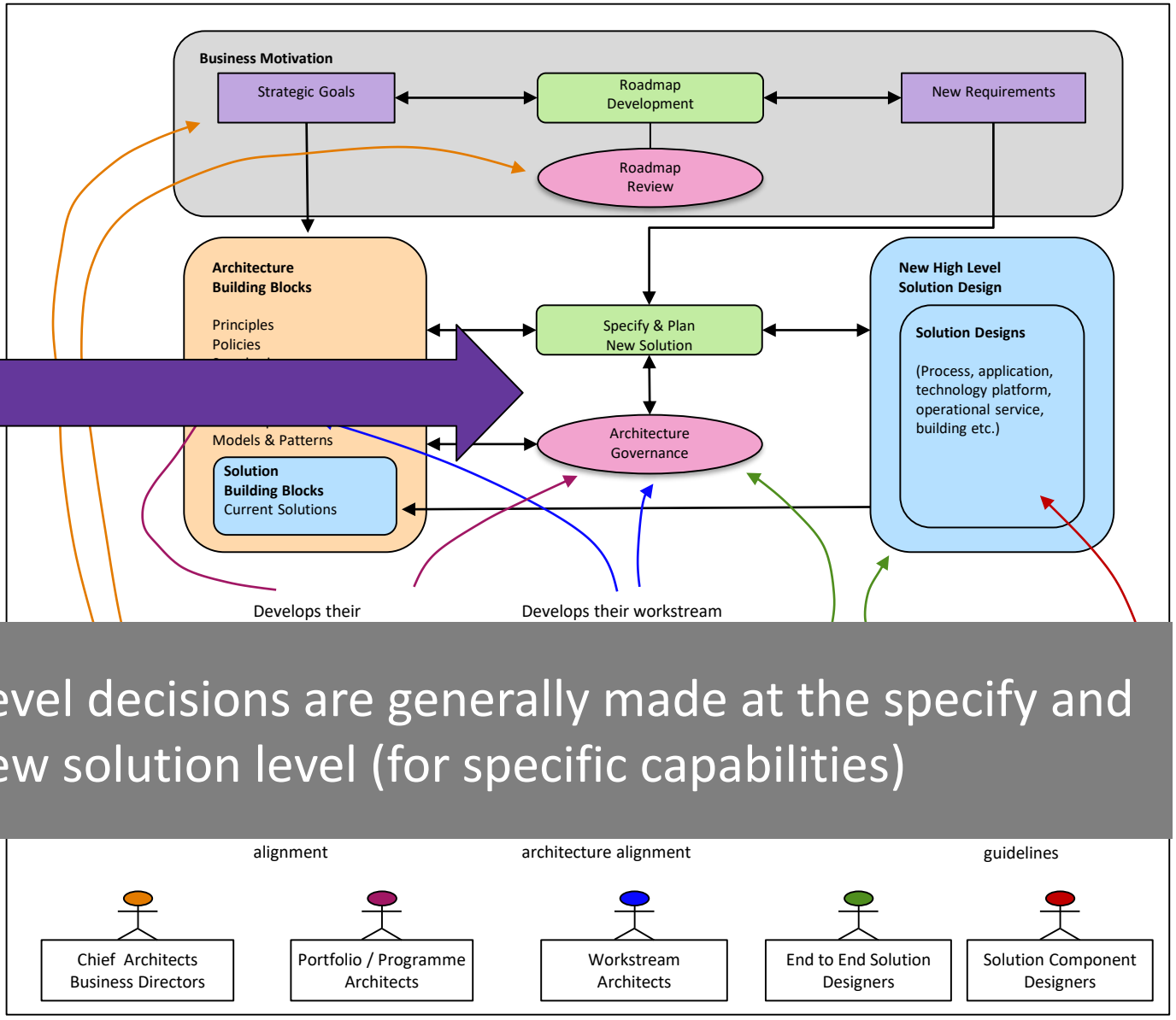
Architecture level decisions are generally made at the roadmap development level (for enterprise / segment / capability)

Example Process For Delivering Architectural Alignment

Delivering change that is aligned to an enterprise architecture is a dynamic process supported by a number of roles.

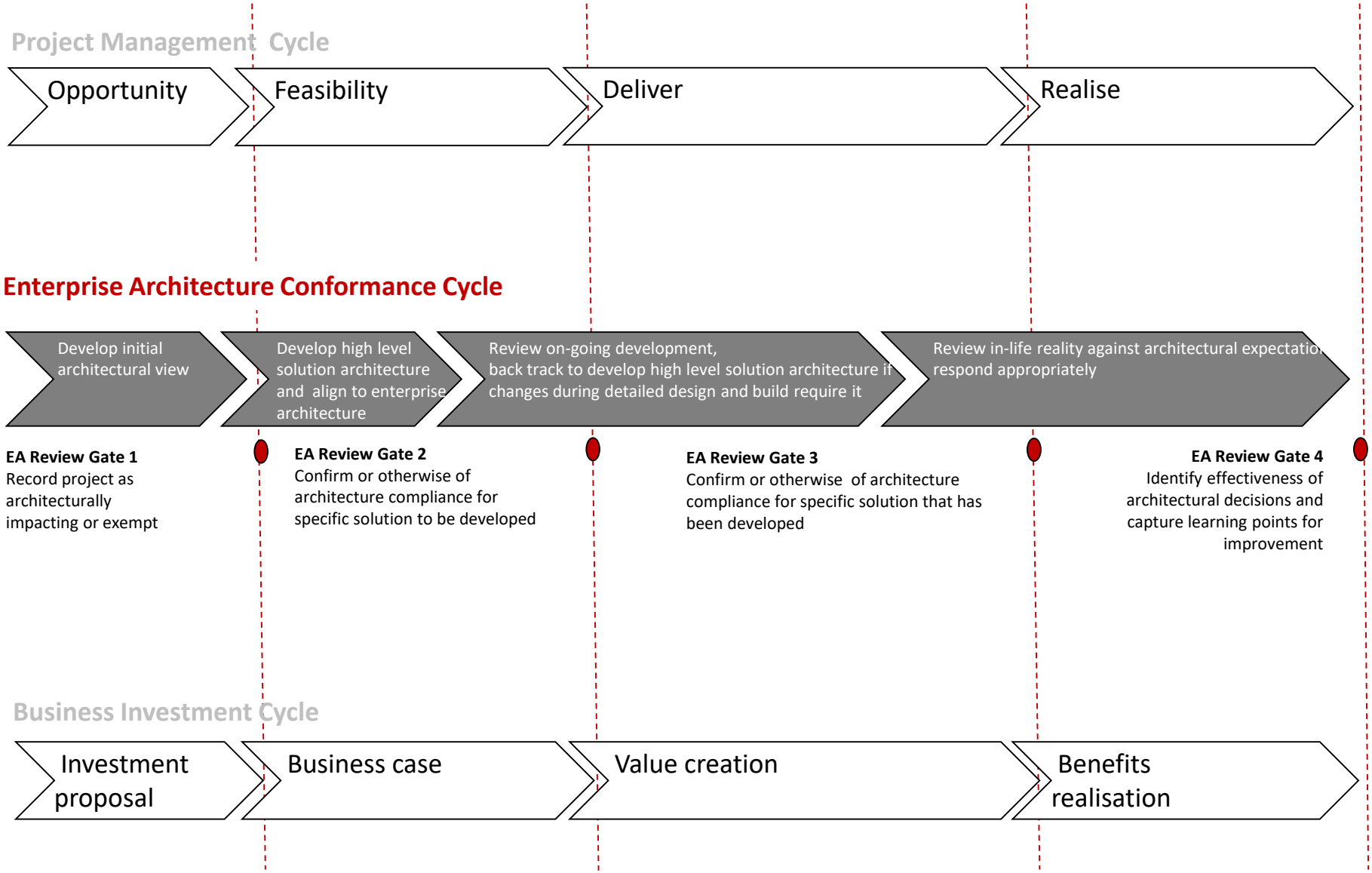
Example roles and their responsibilities are shown.

Architecture governance provides the basis for new and changing architectures and designs to reflect the right balance of architectural conformance and specific requirements for an individual solution.

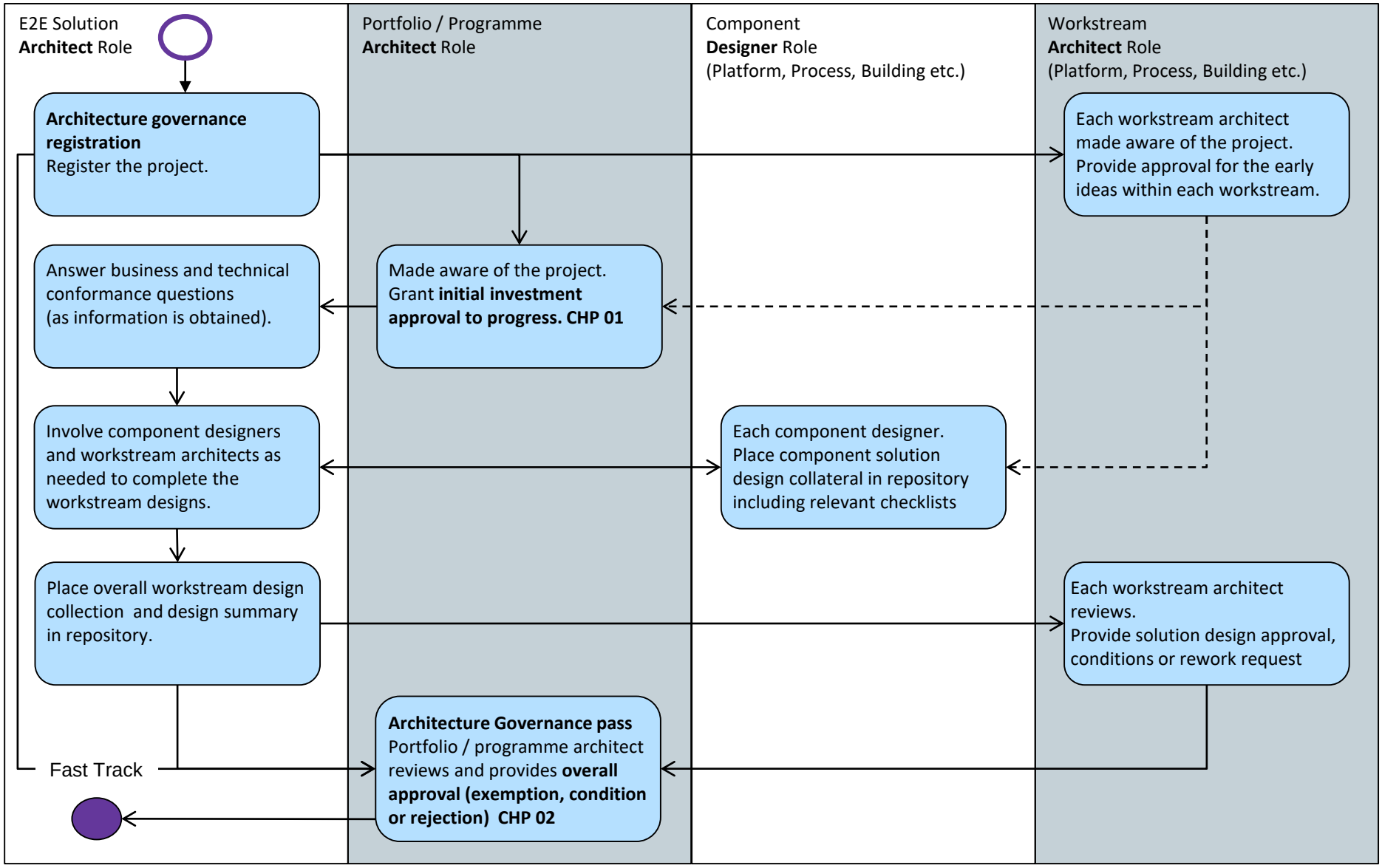


Solution design level decisions are generally made at the specify and plan new solution level (for specific capabilities)

Solution Architecture Conformance Checkpoints



Example Enterprise / Solution Architecture Governance Process



The End

If you would like training and/or consulting services to support your enterprise architecture programme please contact:

Michael Anniss
Greengates, Millbank Street, Nairn
Scotland, IV12 4QA

Tel: +44 (0)7450927005
eMail: mjanniss@aol.com

Website: www.enterprise-architecting.com