

Enterprise Reference Architecture

Addressing key challenges facing EA and enterprise-wide adoption of SOA

Ahmed Fattah

IBM Executive IT Architect

IBM Global Services

Sydney, Australia

Content

- Overview
- Take away points
- Enterprise Architecture (EA), its importance and challenges
- Reference Architecture (RA)
 - RA classification scheme
- Enterprise Reference Architecture (ERA)
 - What is it and how can it help enhance EA?
 - Program-level architecture
 - Why is ERA a natural fit with SOA?
 - ERA lifecycle
- Case studies
 - Case study 1
 - Case study 2
- Summary, conclusions and call for action

The purpose of this presentation is to share my experiences in the use of Reference Architecture for addressing some key EA challenges.

- Problem
 - The importance of EA has been accepted by many organisations, but ...
 - *Huge challenges face many in realising the promised benefits of EA on regular basis.*
- Diagnosis
 - Based on my experience, I observed that a root cause is the *gap/disconnect between EA and project-level architecture.*
 - This gap/disconnect *burdens project architects with the interpretation of EA principles, policies, standards and guidelines* to develop their project architecture. *This is often difficult, time consuming and error prone.*
 - *Similar burden is placed on the Enterprise Architect to police project-level architecture for conformance.*
- Solution
 - The presentation relates my experience in developing and applying an approach that successfully uses **Reference Architecture (RA)** to bridge this gap.
 - This form of RA takes the form of an **Enterprise Reference Architecture (ERA)** which is *a blueprint for the Solution Architecture of a number of potential projects within an organisation that embodies the EA's principles, policies, standards and guidelines.*

ERA can be a very effective tool for enhancing the effectiveness of EA.

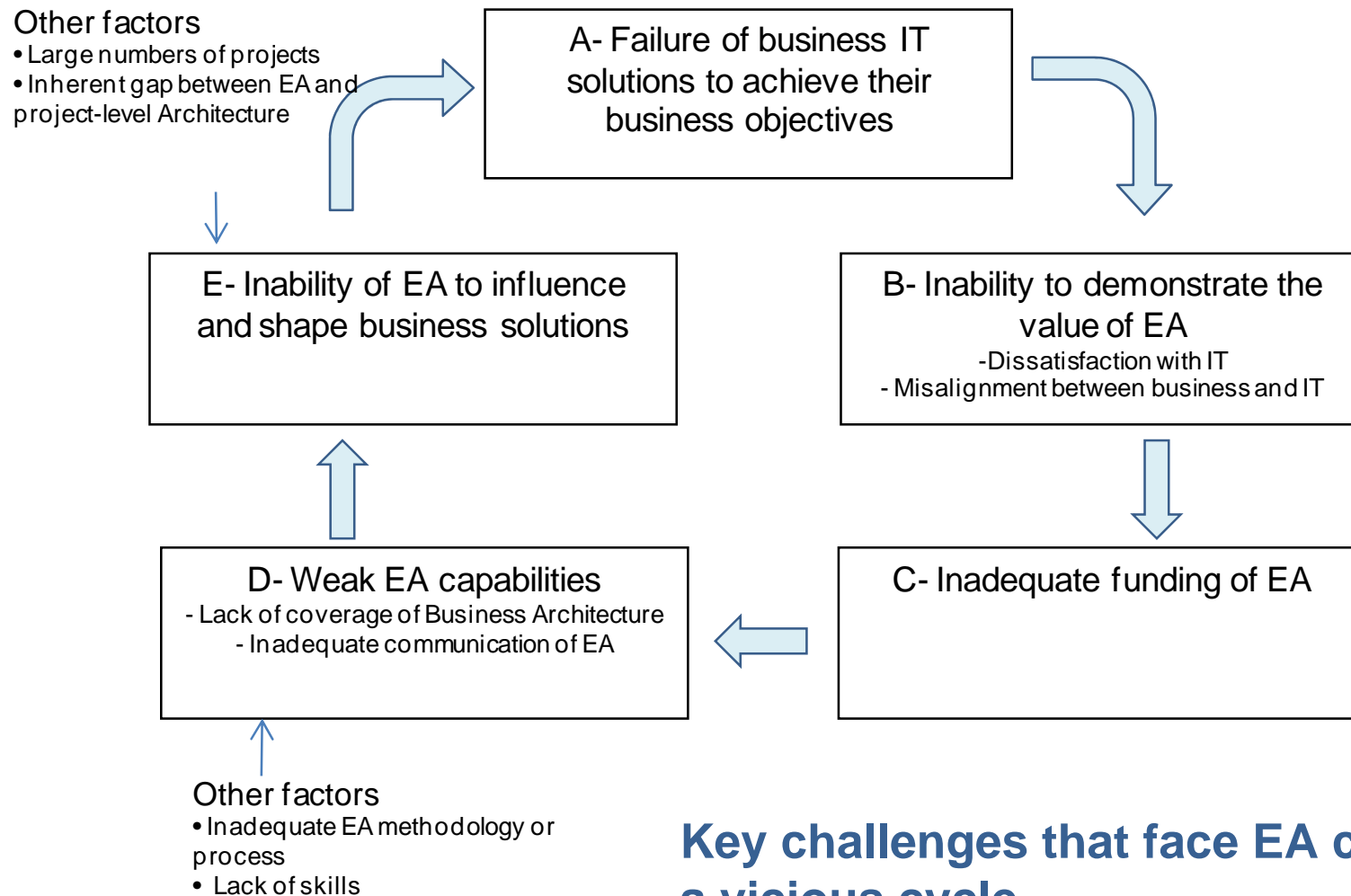
Key take away points of the presentation:

- ERA can help in...
 - **bridging the gap between EA and project-level architecture**

 - **demonstrating the value of EA to the business**
 - **When applied at a program level**

 - **facilitating the enterprise-wide adoption of SOA**

The need for and importance of EA have been accepted by many organisations. However, realising the full potential of EA in many organisations on regular basis is still very challenging.



Key challenges that face EA create a vicious cycle

A root cause behind the challenges facing EA is the wide gap/disconnect between EA and Project-level Architecture.

- This gap/disconnect burdens the project architects to interpret EA principles, policies, standards and guidelines to develop their project architecture. *This is often difficult, time consuming and error prone.*
- Similar burden is placed on the Enterprise Architect to police project-level architecture for conformance which is also *difficult, time consuming and always controversial.*
- This leads projects to resist or ignore EA partially or completely and to a sense of hostility between Enterprise Architects and projects.
- One reason for the wide gap between EA and Project-level Architecture is that although they share the term architecture, they are practiced and documented very differently.
- This is not surprising since the two architectures serve different purposes.
 - EA primarily takes the form of principles, policies, standards and guidelines.
 - Project-level Architecture takes the form of Architectural Decisions, components, interfaces and their relationships.

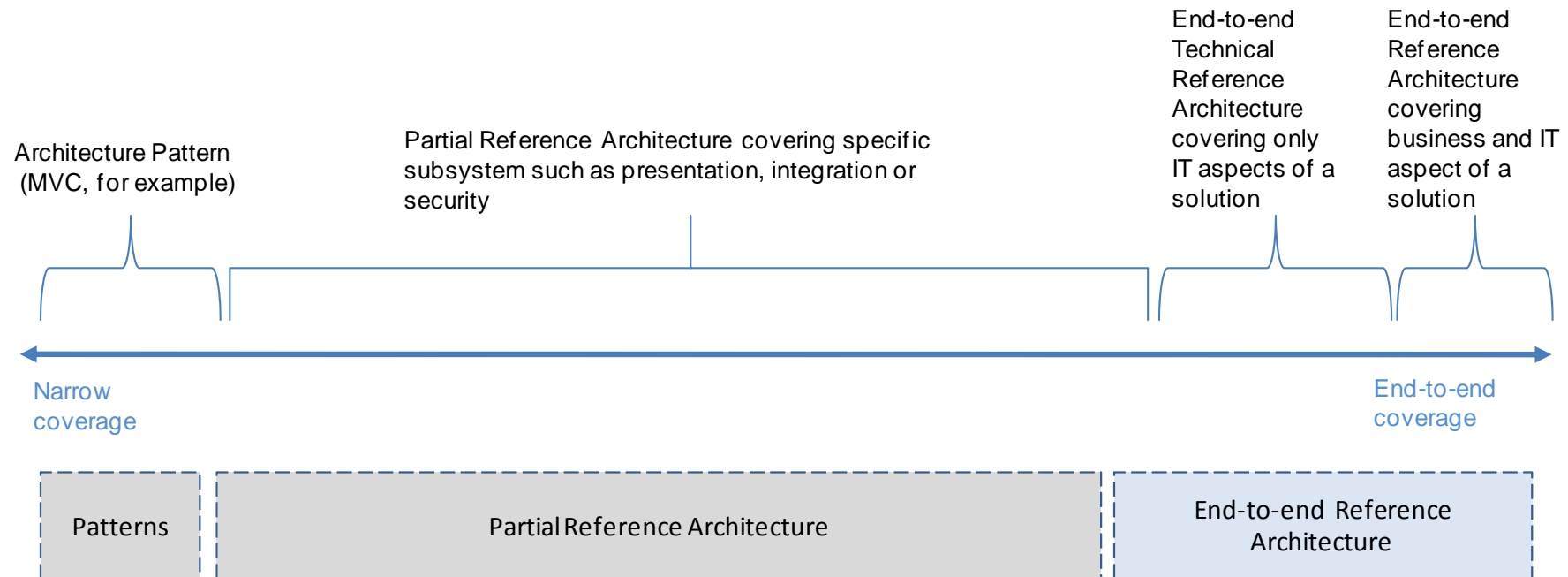
The term Reference Architecture (RA) is used in the industry to refer to a wide variety of constructs from high level abstract conceptual models to a specialised technical architecture.

- There are many definitions of Reference Architecture that although slightly different, have many common elements.
- For our purpose here, the Wikipedia's definition is adequate:
 - **“A Reference Architecture provides a proven template solution for an architecture for a particular domain. It also provides a common vocabulary with which to discuss implementations, often with the aim to stress commonality”.**
- Examples of RA cover a very wide range:
 - Unisys 3D Blueprint [5] which covers strategy, business processes, applications and infrastructure.
 - Specialised technical architecture such as IBM WebSphere Integration Reference Architecture [3].
- The terms *Reference Architecture* and *Reference Model* are sometimes used interchangeably.

Although I have used Reference Architectures for a long time, I was surprised when I reviewed the staggering range of usage of the term recently.

- Google search for the term “Reference Architecture” has over *300,000* hits
- I first assumed that some of these usages must be erroneous. However, I realised that this was not the case, at least for the many instances I have surveyed...
- But that the culprit is the malleability of the term architecture itself.
- This means that anything you can think of can have an architecture and by extension a RA.
- My thesis is based on the belief that Reference Architectures can be distinguished along two dimensions:
 - *Coverage*
 - *Level of abstraction*

Coverage or applicability indicates the area where a RA can be useful. Some RAs cover only presentation, integration or security aspects of solutions, others cover an end-to-end enterprise solution.

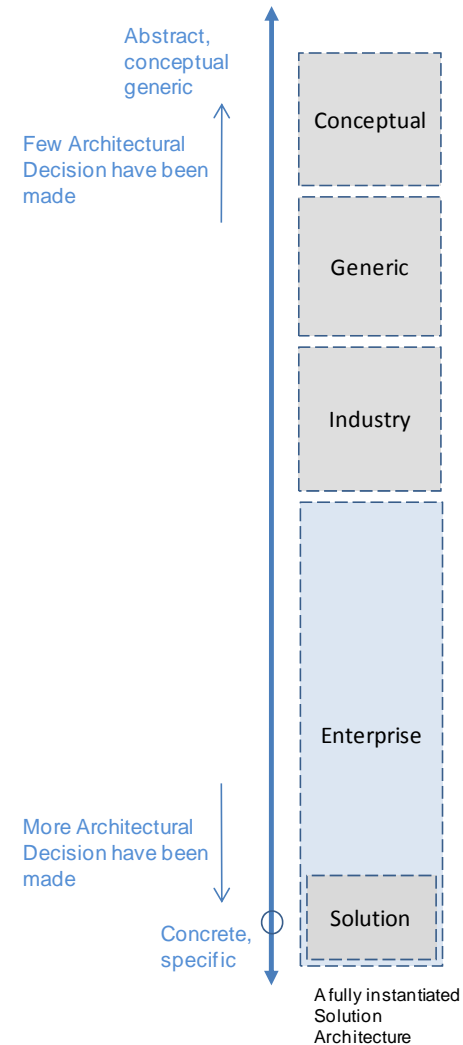


Level of Abstraction reflects how concrete or specific a given RA is. It indicates ultimately the gap between the RA and a Solution Architecture based on it.

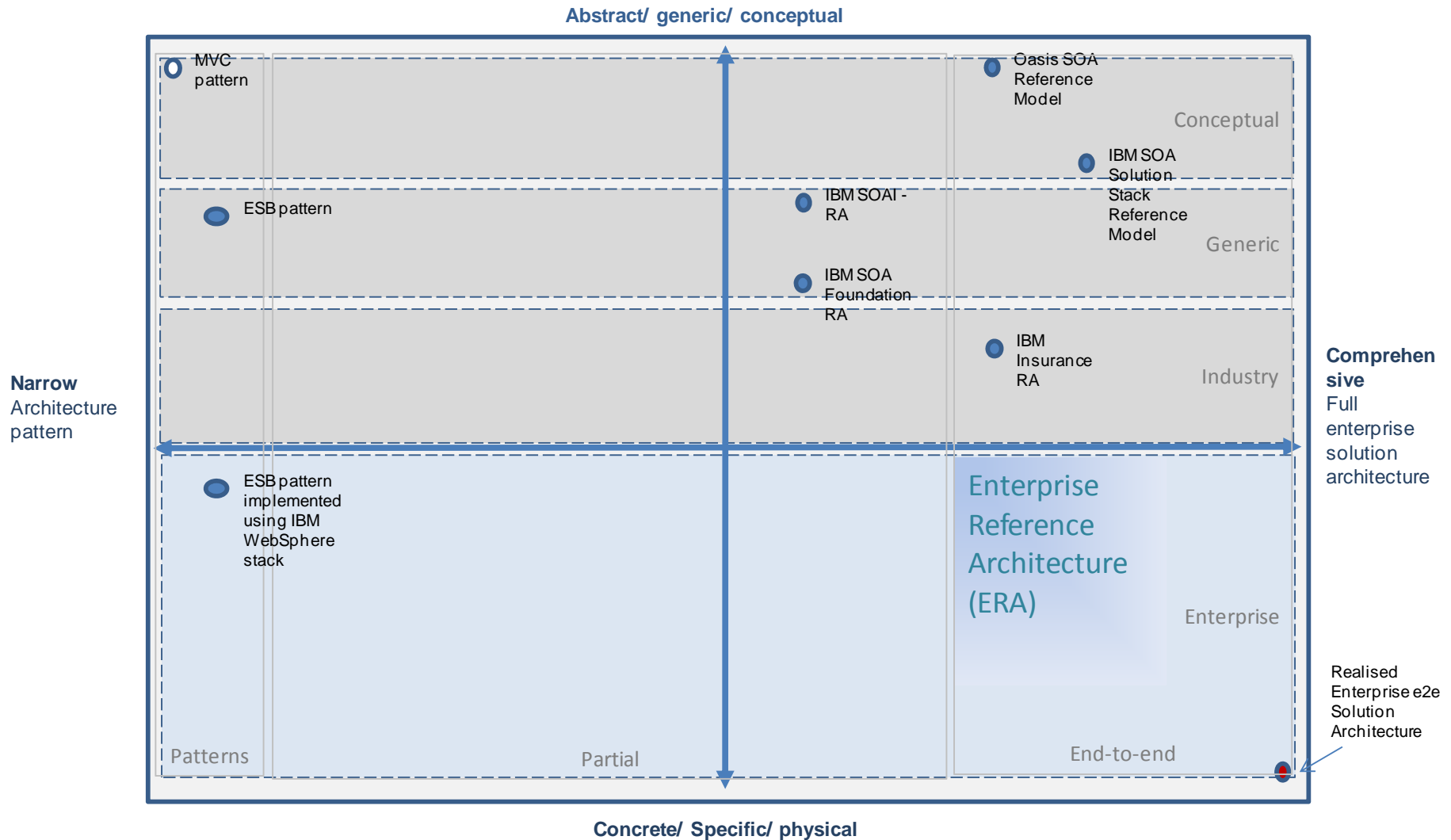
Reference Architecture can be defined at varying levels of abstraction from the conceptual and generic to the concrete and specific

Another useful way to think about this dimension is in terms of Architectural Decisions.

On the concrete/specific end, 'all' the Architectural Decisions have been made. On the other end, very few Architectural Decisions are likely to have been made.



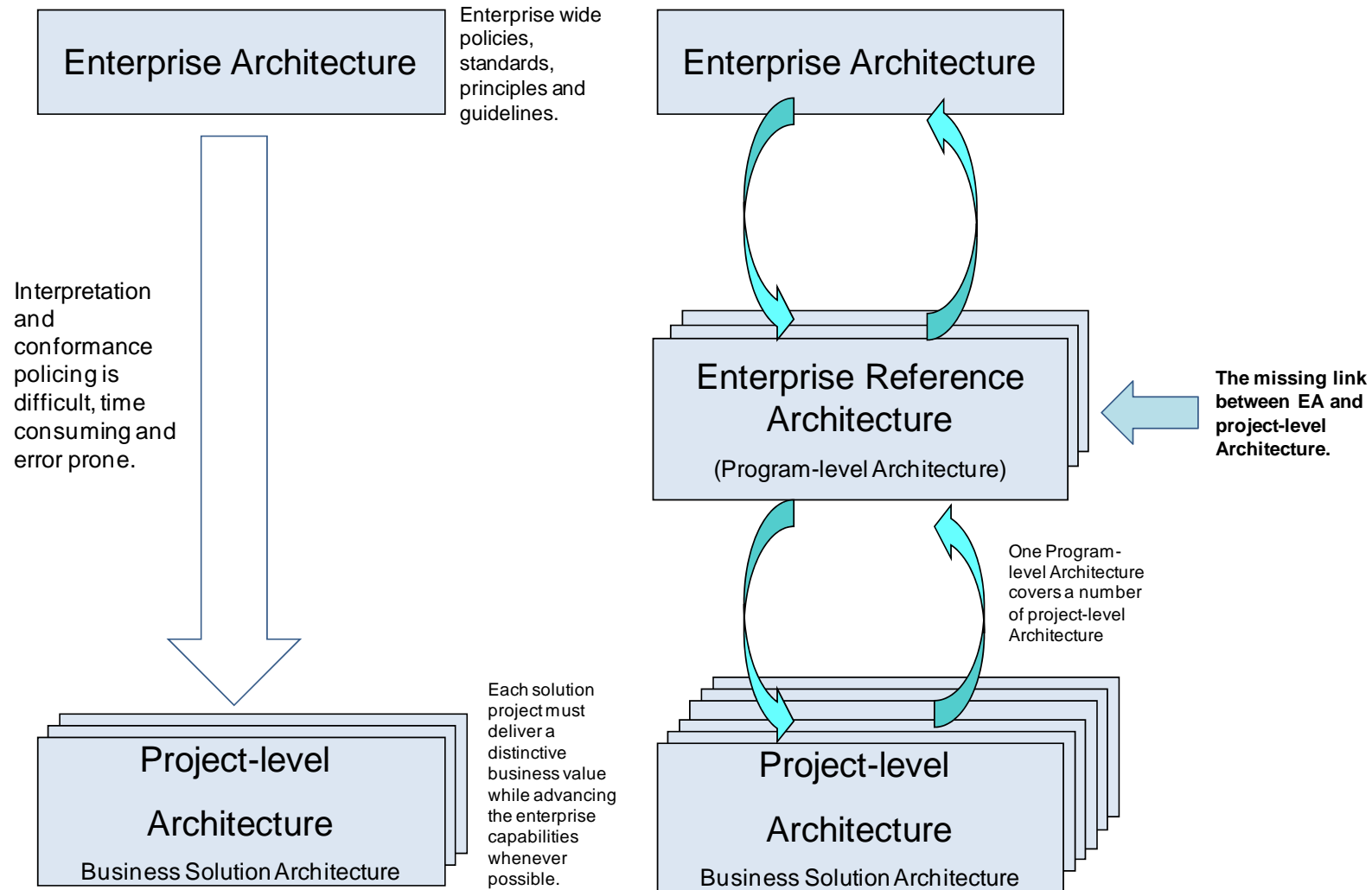
The classification scheme is useful in sorting out the myriad of RAs that are available to determine which are useful in a given situation and how different RA are related to each other.



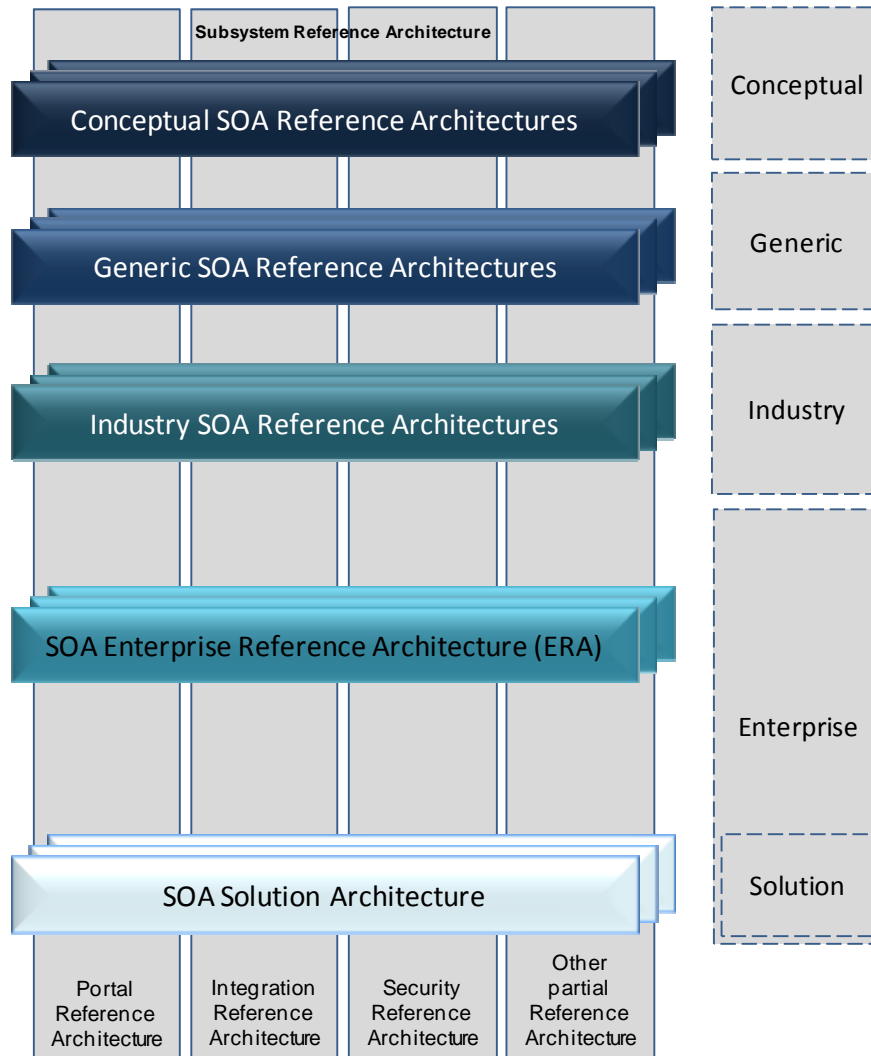
An ERA is a blueprint for the Solution Architecture of a number of potential projects within an organisation that embodies the EA principles, policies, standards and guidelines.

- An ERA is a Solution Architecture with some of the *Architectural Decisions* being made and others left open.
- ERAs resemble actual Solution Architectures. This means that the effort to apply them by project-level architects is relatively low.
- They are applicable to a number of potential business solutions within the organisation.
- Ideally, the development of ERA should be funded directly by the business to address specific business objectives.
- One key source of knowledge, experience and reusable components for the development and construction of ERAs must come from existing projects by way of harvesting suitable proven components and patterns.

Funding the development of an ERA directly by the business for a specific business initiative (a program of projects) can profoundly transform the way architecture is practiced in the organisation. I refer to this as Program-level Architecture.

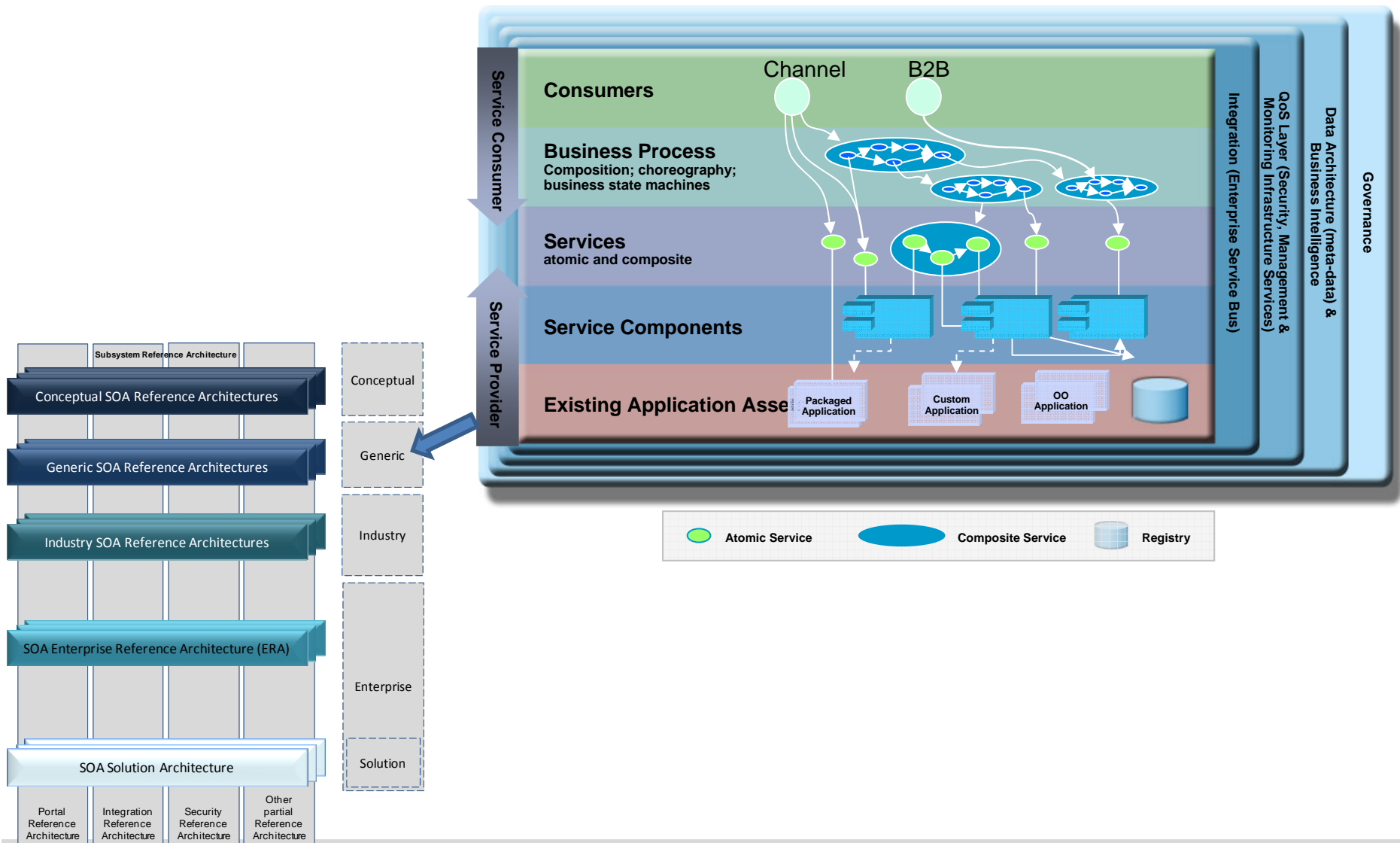


Although the ERA approach proposed in this paper applies to all architecture styles, it's more compelling and easier to apply for SOA because of SOA's emphasis on standardisation and reuse.

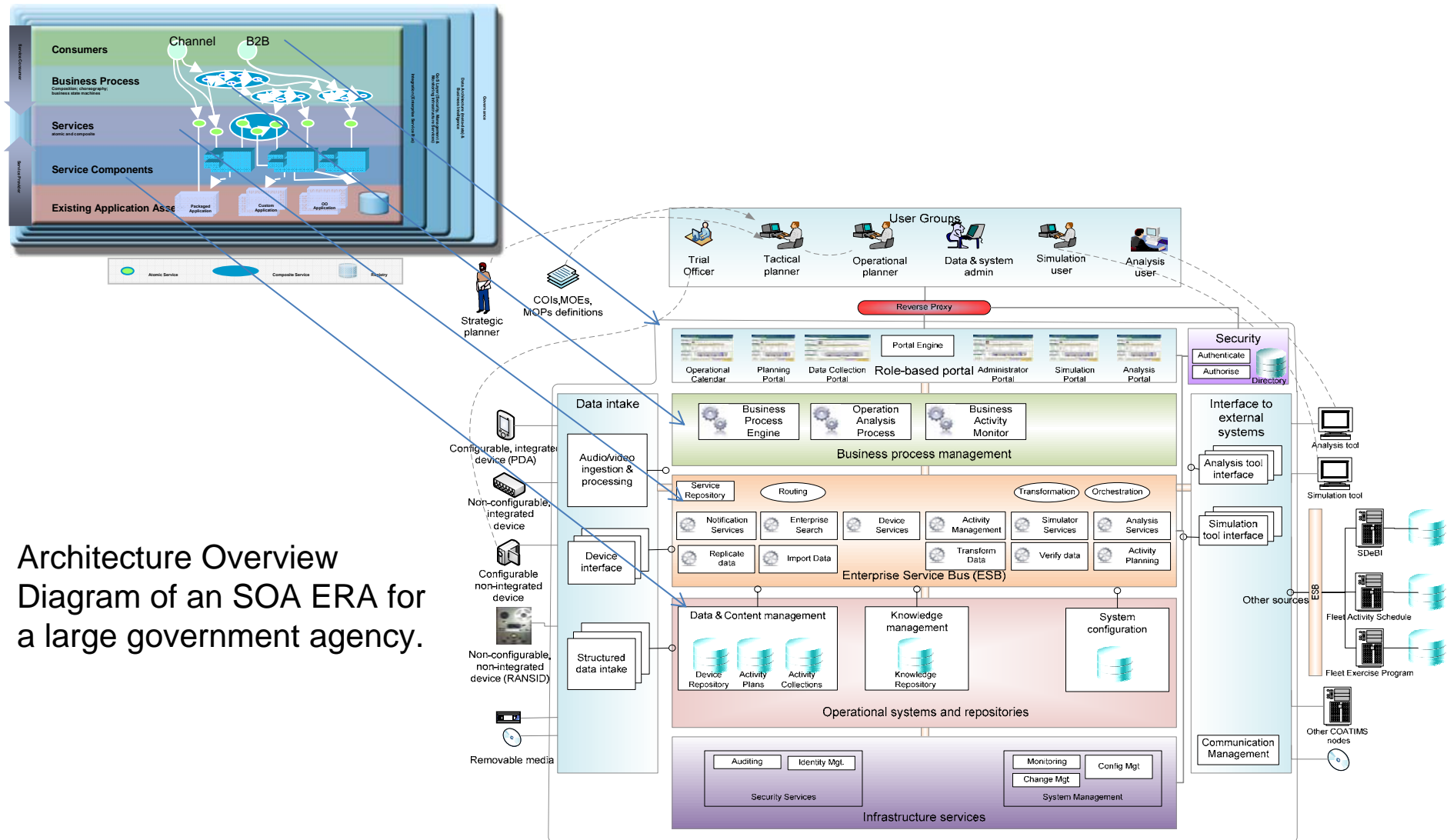


The hierarchy of the SOA RAs that can be adopted and applied within an organisation culminating in a small number of ERAs that can be used to guide projects in creating SOA business solutions.

IBM SOA Solution Stack (S3) Reference Architecture is invaluable for creating an ERA for most of the operational business solutions for an enterprise.

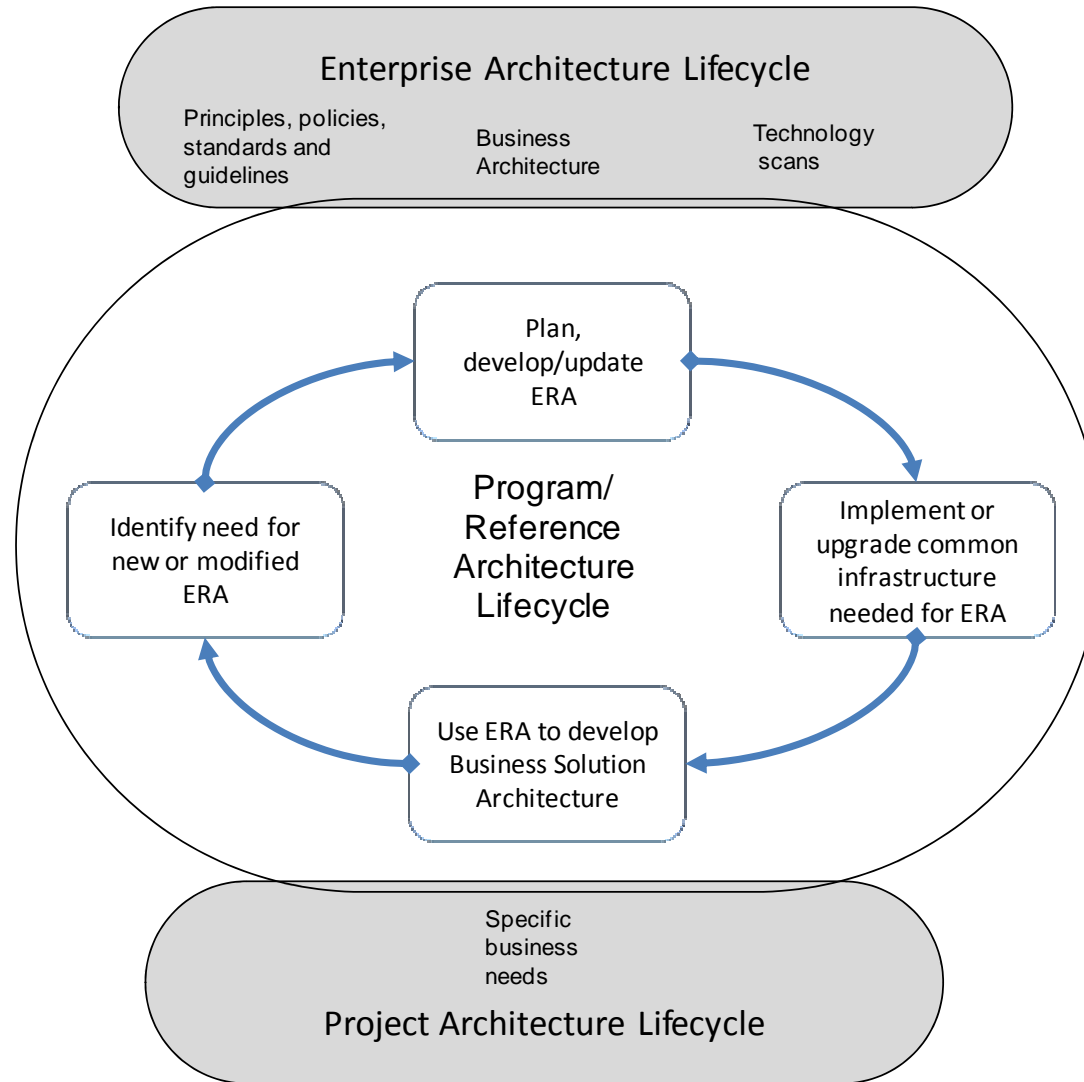


Developing an SOA ERA based on the IBM S3 RA can be done systematically by mapping each layer to technical and functional components.



Architecture Overview
Diagram of an SOA ERA for a large government agency.

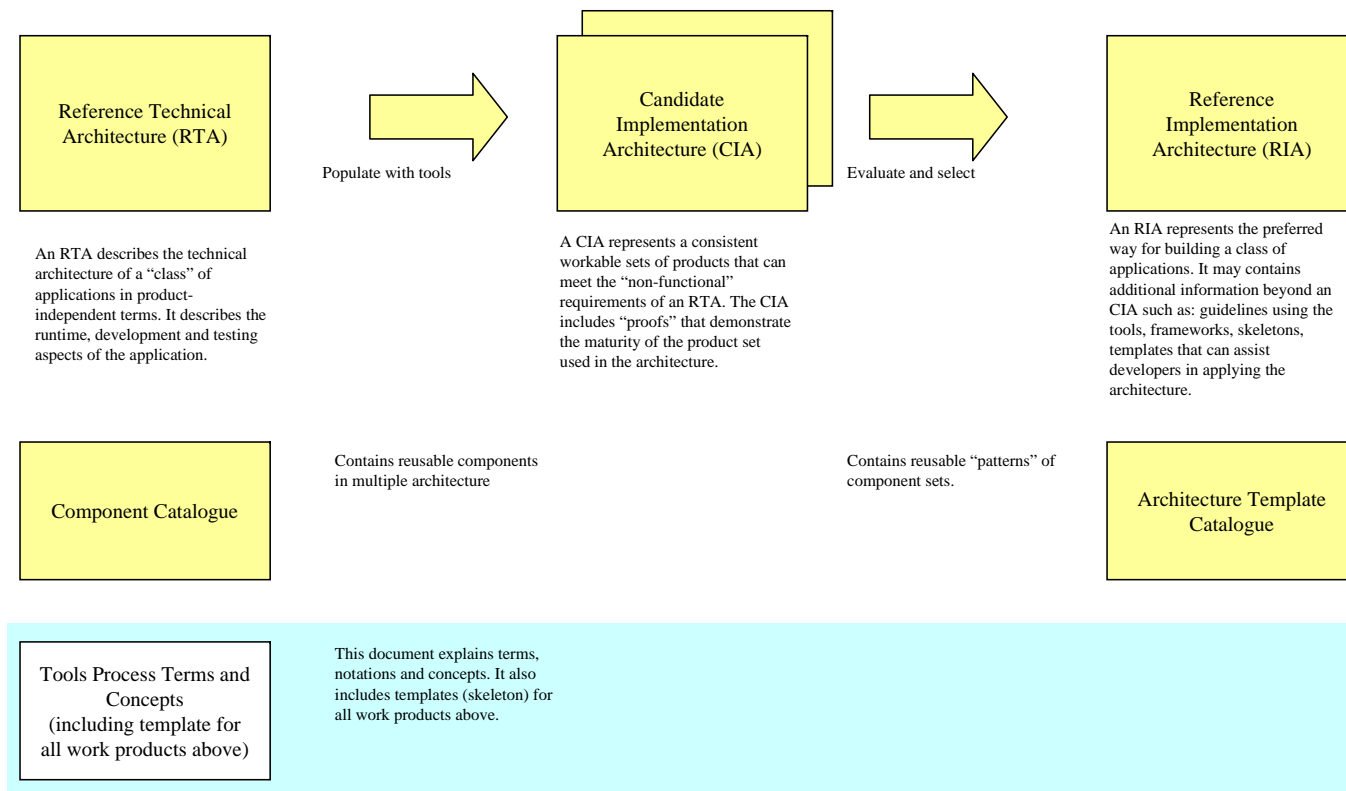
The process for developing and using ERA overlaps with the lifecycle of EA and projects and should be compatible with most typical EA and software development lifecycle methods and processes.



Case study 1 used the ERA approach to address the problem of the lack of standardisation of application platforms, databases, middleware, development tools etc in a large organisation.

- The EA group was well funded compared with many other organisations, the group's resources were stretched in maintaining the EA guidance artefacts and 'policing' the conformance of solutions.
- EA guidance was contained in very large volumes that covered mainly two categories:
 - SOE (Standard Operating Environment) which covers standards concerning what development platforms, middleware, etc are allowed to be used; and
 - Architectural guiding principles that should be followed by the projects when developing their architecture.
- We found that this material was well written and maintained but also found some problems with the way the EA guidance was provided especially from projects' point of view:
 - One of the problems that projects were faced with in interpreting the SOE was the compatibility between choices of various solution components.
 - Interpreting the architectural guiding principles was a problem because many of these principles (and there were many) conflict and the degree of required conformity varied from *must conform* to *nice to have*.
 - The EA group got many requests for clarification and exemption from adhering to the SOE or the guiding principles.
 - Projects did not view the EA group as a help but an obstacle that they have to go around.

A number of ERAs were developed and adapted in many projects and the approach in general helped greatly in revitalising the EA of the enterprise.



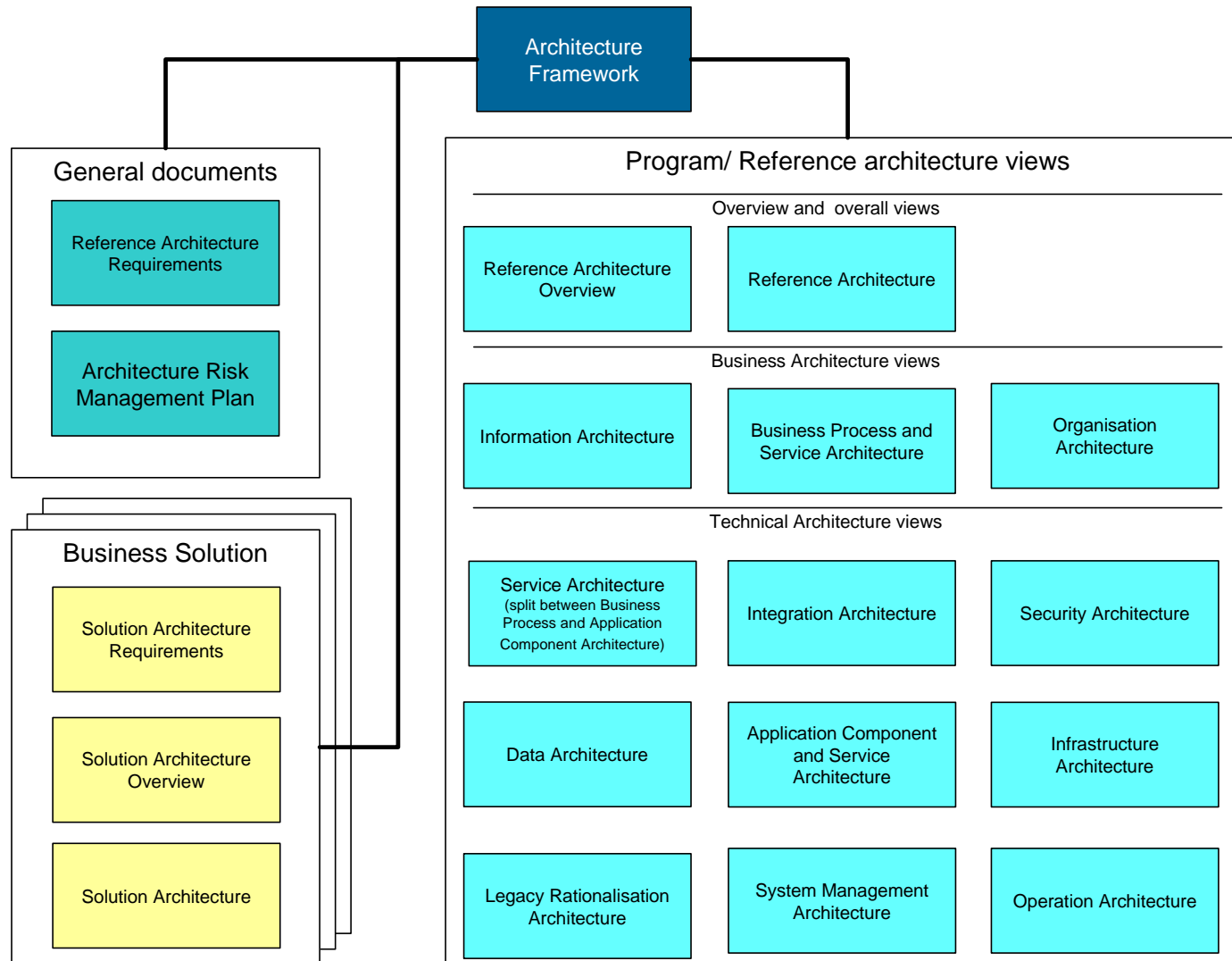
One of the lessons learned was that even in very large organisations, the most common types of business solutions can be covered with very few types of Reference Architecture especially the platform-independent type (RTA).

- Other lessons learned include:
 - Funding the development of ERAs can be difficult to obtain without tying them to specific business initiatives. After the initial development of a number of ERAs the momentum slowed down because of a lack of funding.
 - Another lessons that is also related to the funding of the ERAs is the problem of the proliferation of types of ERAs which makes identifying suitable ERAs for a given business situation difficult.
- Other details of the case study are available on request

In this case study we developed and applied most of the concepts behind the ERA approach including the funding of the development of ERA.

- The context of this case study was a large government department that embarked on a massive transformation program.
- The advice to the CIO was to ensure that this new platform is used in an enterprise-wide manner and not on a project by project basis. The CIO approached us and asked *“but how can I do this while I have many lines of business requesting to start developing a number of individual business solutions immediately using the new platform?”*
- The answer was to defer the start of these projects until an ERA was developed to guide how these projects were developed and maximise the potential level of reuse between these projects.
- So the development of this ERA was funded explicitly with the aim to lower the overall cost of the new business solutions.

One important aspect of this case study that was crucial but difficult to achieve was the inclusion of the Business Architecture in the development of the ERA.



One of the key lessons learned was that the development of ERA can significantly reduce the effort and risk associated with development of individual projects.

- Other lessons learned were:
 - The Solution Architects who work on the development of the ERA can contribute enormously to the projects in their beginning.
 - It is not enough to just initially develop the ERA. The architecture should be maintained and enhanced by lesson learned from individual projects. This means that some architectural resources must be allocated to maintaining the ERA. Ideally these resources are assigned on rotation basis to the development projects and the EA group.
 - Developing ERAs is not a substitute for typical EA activities. Although knowledge and information from the ERAs can feed back to other EA activities, there is a need to address other needs within the organisation that ERAs do not cover.
- Other details of the case study are available on request

The presentation discussed an approach for enhancing the effectiveness of EA practices with the use of Enterprise Reference Architecture (ERA).

- ERAs are Reference Architectures that embody the principles, policies, standards and guidelines of the enterprise in a form that is easily applicable to a set of business solutions. ERAs are ideally developed for large business initiatives.
- The approach described here was developed and has been applied successfully in a number of practical situations.
- I hope that some of the audience find the whole approach or some aspects of it useful for their organisations or clients.
- I welcome all feedback regarding the structure, contents or experiences related to applying the concepts discussed in the presentation.
- I aim to enhance the approach and the concept of ERA based on feedback and from a number of customer situations that I am currently involved in.

धन्यवाद

Hindi

多謝

Traditional Chinese

ขอบคุณ

Thai

Спасибо

Russian

Gracias

Spanish

Thank

English

شكراً

Arabic

You

Merci

French

Obrigado

Brazilian Portuguese

Grazie

Italian

多谢

Simplified Chinese

Danke

German

நன்றி

Tamil

ありがとうございました

Japanese

감사합니다

Korean

Supporting slides

End of presentation