

# 1. Anatomy of the e-Framework technical model

## Supported by a knowledge base

The technical model of the e-Framework aims to facilitate system interoperability via a service-oriented approach. The model provides:

- Knowledge base (or assets register) of standards-based interfaces to education and research systems
- Vocabulary for consistently modelling services as technical components
- Method for describing how processes can be supported by those services
- Guidance on how to use the components

## Key concepts and components

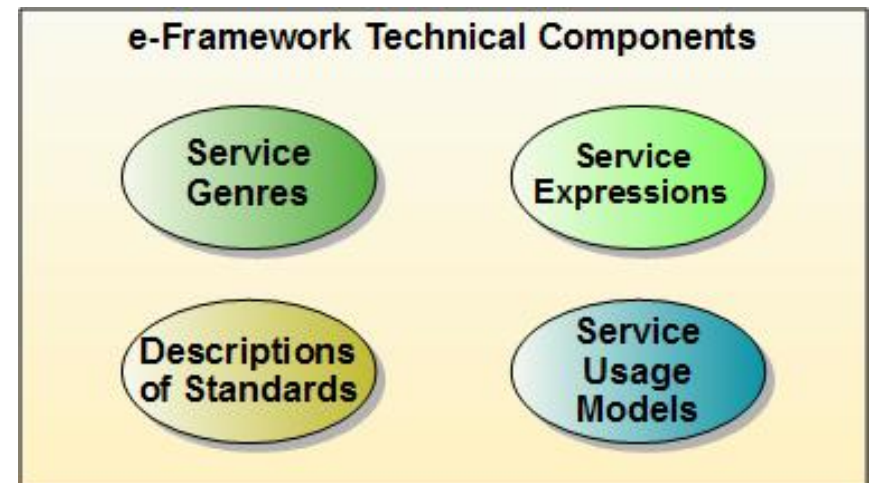
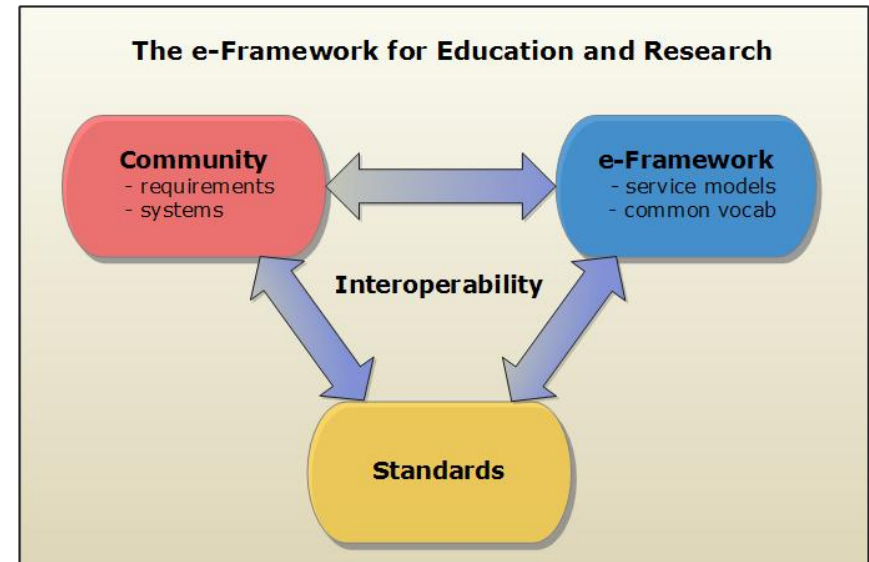
Technical components used to explain the e-Framework concepts include:

- Service Genres – descriptions of the range of behaviours exhibited by services
- Service Expressions – descriptions of how the interfaces to a service work
- e-Framework Descriptions of Standards – information about a standard that is relevant to the e-Framework conceptual model
- Service Usage Models (SUMs) – descriptions of how the other components are combined to meet a specific purpose

The *NEXT* diagram in this series shows how these technical components are related within the model.

Complete set of explanatory diagrams:

1. [Key technical components](#)
2. [Relationships in the model](#)
3. [What's in a SUM](#)
4. [Developing a SUM](#)
5. [Community engagement](#)
6. [Building the e-Framework](#)
7. [Bringing it all together](#)
8. [e-Framework and standards](#)



## 2. Relationships between Service Genres and Service Expressions

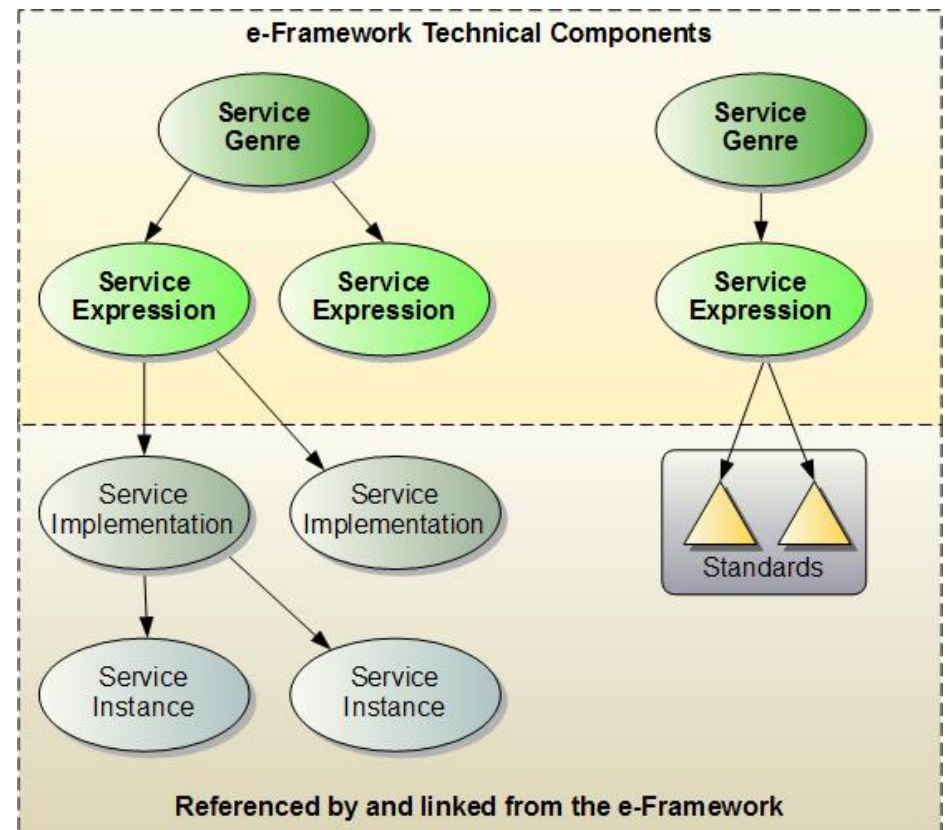
### How service genres and service expressions are related to actual deployed services

- Service Genres are technology-neutral descriptions of the behaviours of services
- Service Genres can be bound to specific technologies by one or more Service Expressions; Service Genres can also be abstracted from Service Expressions
- Service Expressions can be implemented in more than one way as Service Implementations, and these implementations can be deployed in more than one place as Service Instances
- [Standards](#) provide the interoperability of the data and messages used in the services
- Service Implementations and Instances may be referenced by the e-Framework

The [NEXT](#) diagram in this series shows how collections of services (SUMs) bridge between business needs and IT infrastructure.

Complete set of explanatory diagrams:

1. [Key technical components](#)
2. [Relationships in the model](#)
3. [What's in a SUM](#)
4. [Developing a SUM](#)
5. [Community engagement](#)
6. [Building the e-Framework](#)
7. [Bringing it all together](#)
8. [e-Framework and standards](#)



### 3. Service Usage Models (SUMs) – bridging business & IT infrastructure

#### Collections of services called SUMs lead to business solutions

SUMs are composed of either Service Genres or Service Expressions, but not a mixture:

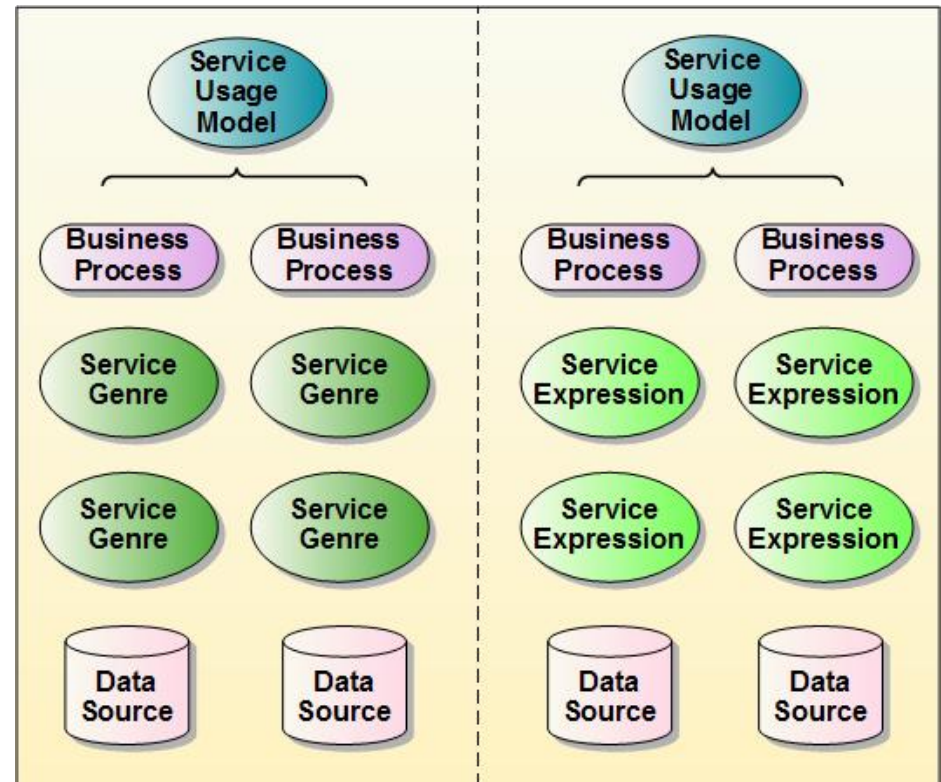
- SUMs can be composed of two or more Service Genres (left panel)
- SUMs can be composed of two or more Service Expressions (right panel)

SUMs also describe:

- Data sources they use
- Business processes that they support

Placeholders can be used for missing components in a SUM, but these should be developed.

The [NEXT](#) diagram in this series shows how an example business process can be modelled as a SUM.



Complete set of explanatory diagrams:

1. [Key technical components](#)
2. [Relationships in the model](#)
3. [What's in a SUM](#)
4. [Developing a SUM](#)
5. [Community engagement](#)
6. [Building the e-Framework](#)
7. [Bringing it all together](#)
8. [e-Framework and standards](#)

## 4. Developing a SUM for student course registration

### Service Usage Models (SUMs) capture real activities

Registering a student for a course is a real-world process that can be modelled as a SUM.

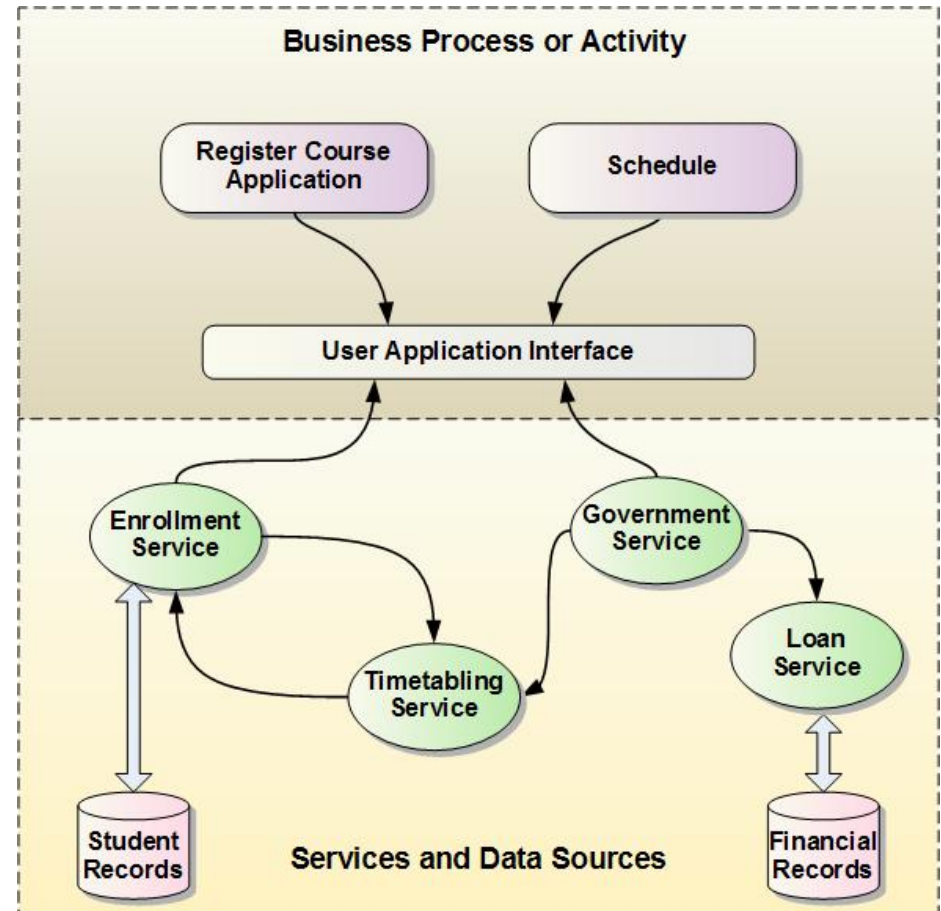
The SUM documentation:

- References the business processes and activities that motivated the development of the SUM even though these are not components of the e-Framework model for describing a SUM
- Translates business terminology into e-Framework technical component vocabulary
- Describes how the process-specific technical components and data sources within the SUM are organised behind the user-application interface

The *NEXT* diagram in this series shows how the community contributes to the evolution of the technical components.

Complete set of explanatory diagrams:

1. [Key technical components](#)
2. [Relationships in the model](#)
3. [What's in a SUM](#)
4. [Developing a SUM](#)
5. [Community engagement](#)
6. [Building the e-Framework](#)
7. [Bringing it all together](#)
8. [e-Framework and standards](#)



## 5. Community engagement – a key part of the model

### Contributing and re-using components - an iterative and evolving process

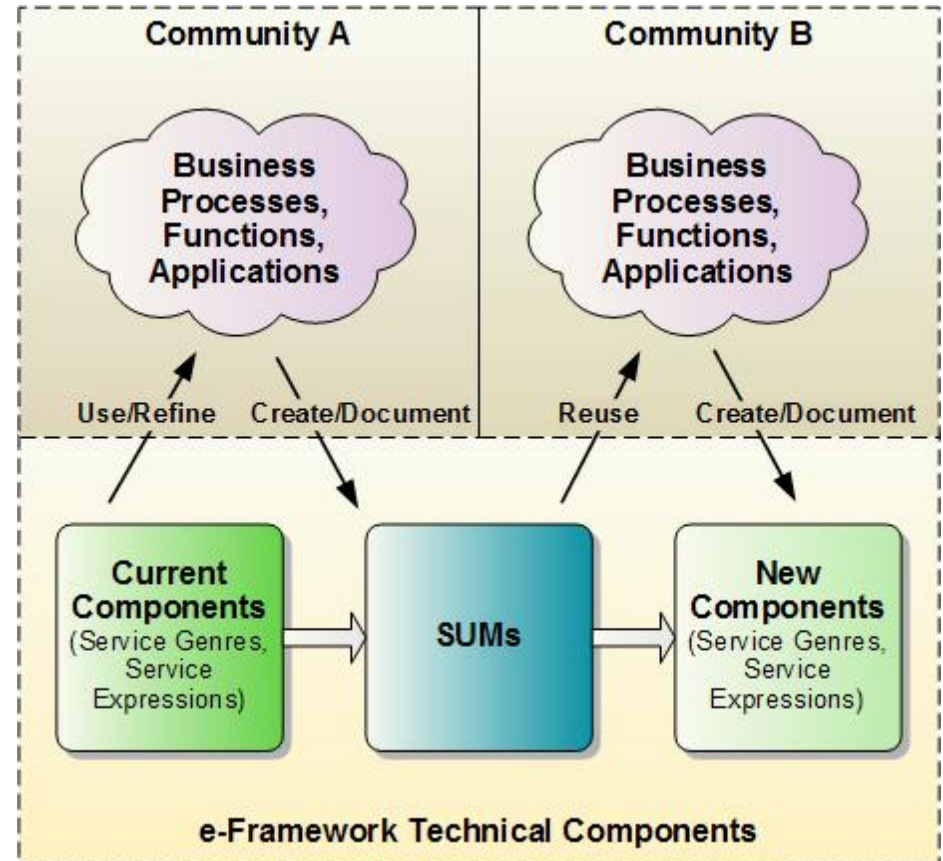
The service modelling within a community can benefit other communities when they share their experiences via the e-Framework. The following is a typical scenario:

- Community members analyse their business processes, identify functions, and develop applications
- Communities also explore the e-Framework registries to find components that they can use and adapt
- Communities document the collection of services they developed as e-Framework Service Usage Models (SUMs)
- SUMs will typically re-use services from the existing registry of technical components, but documenting a SUM may also lead to the development of new Service Genres and Service Expressions
- SUMs and other components can be used/refined by the original community or adopted/reused by other communities

The [NEXT](#) diagram in this series shows how feedback from built systems aids cross-fertilisation between communities.

Complete set of explanatory diagrams:

1. [Key technical components](#)
2. [Relationships in the model](#)
3. [What's in a SUM](#)
4. [Developing a SUM](#)
5. [Community engagement](#)
6. [Building the e-Framework](#)
7. [Bringing it all together](#)
8. [e-Framework and standards](#)



## 6. Building the e-Framework

### Cross-fertilisation between communities

The e-Framework evolves based on the feedback and experiences of the communities that use it.

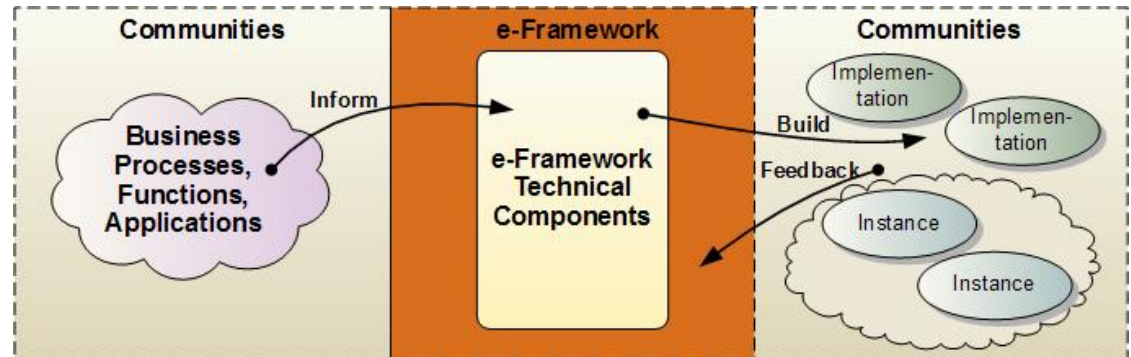
Community input provides:

- Feedback into the e-Framework as refinements for the technical components
- Vital information for supporting guides
- New connections between communities

The [NEXT](#) diagram in this series shows how community activities, standard and the e-Framework model facilitate interoperability.

Complete set of explanatory diagrams:

1. [Key technical components](#)
2. [Relationships in the model](#)
3. [What's in a SUM](#)
4. [Developing a SUM](#)
5. [Community engagement](#)
6. [Building the e-Framework](#)
7. [Bringing it all together](#)
8. [e-Framework and standards](#)



## 7. Bringing it all together to facilitate interoperability – the model, standards and community activities

**Inform, use, build, deploy, feedback, refine, repeat . . .**

The e-Framework advocates the use of standards to benefit education and research in everyday work, facilitating interoperability between commonly used systems.

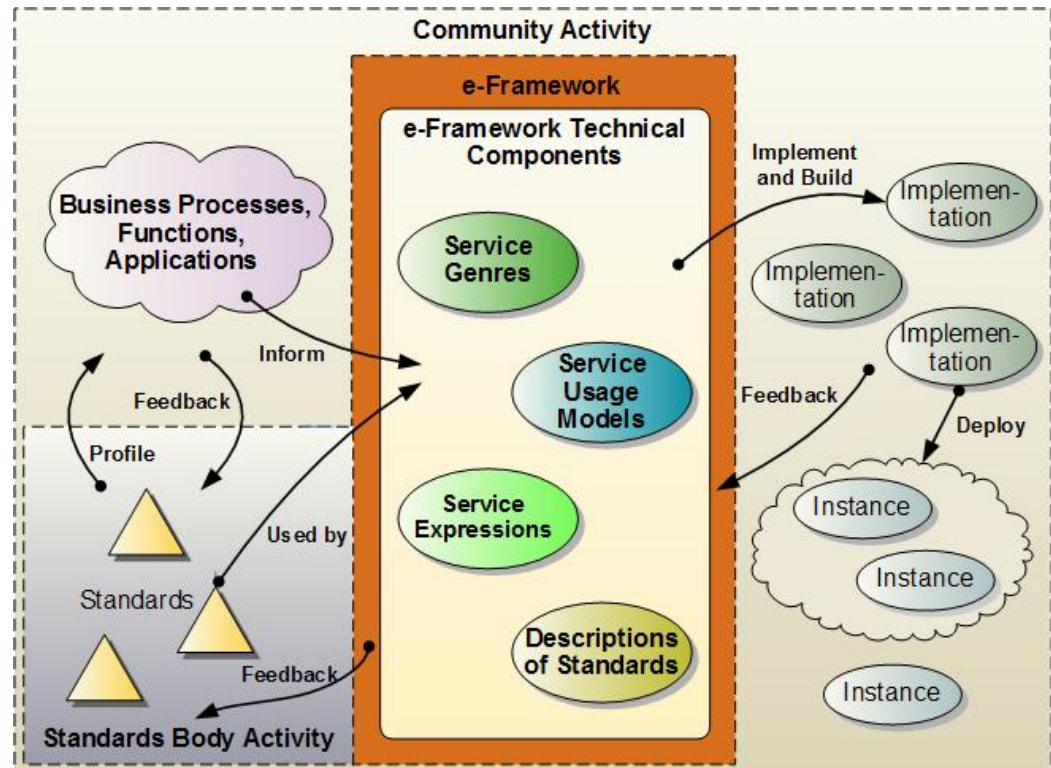
Ongoing interactions between the e-Framework, standards bodies and communities enable further refinement of the technical components. This leads to continual, incremental development and ultimately to improved interoperability.

The diagram recaps the main points:

- [Standards](#) and specifications are used in the various e-Framework components
- Communities build on and implement applications from these components, promoting the use of standards
- Experiences of the community are fed back to the e-Framework and standards bodies, allowing development and refinement cycles to continue
- Interoperability is facilitated

*Re-visit other explanatory diagrams for more details.*

1. [Key technical components](#)
2. [Relationships in the model](#)
3. [What's in a SUM](#)
4. [Developing a SUM](#)
5. [Community engagement](#)
6. [Building the e-Framework](#)
7. [Bringing it all together](#)
8. [e-Framework and standards](#)



## 8. The role of standards in the technical model

**Standards provide the starting point for interoperability of data and messages used within services**

- Standards bodies develop protocols and schemas
- Service Genres document the request-and-response behaviour of the services they describe
- Service Expressions document how those behaviours are implemented in terms of requests, responses and messages for services that are used in a particular context or community
- Service Expressions also document the specifics of how a community uses the protocols and schemas developed by standards bodies and communities

### Summary

- Service Expressions facilitate the interoperability of implemented and deployed services

*Complete set of explanatory diagrams:*

1. [Key technical components](#)
2. [Relationships in the model](#)
3. [What's in a SUM](#)
4. [Developing a SUM](#)
5. [Community engagement](#)
6. [Building the e-Framework](#)
7. [Bringing it all together](#)
8. [e-Framework and standards](#)

