The e-Framework for Education and Research

Service Classification Scheme

March 2007

Introduction

This document presents recommendations for describing the service classification facets used to classify the core components (service usage models, service genres, and service expressions) of the e-Framework. These conventions apply to the official e-Framework published components. Submitters and other communities may use their own additional classifiers.

> **NB:** Develop some examples, one for each type of e-Framework component. The examples should be in separate documents.

A collection of explanatory notes are distributed throughout the document tagged NB. Recommendations to the stakeholders are presented as notes tagged REC.

The words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in [RFC 2119].

Stakeholders and Audience

Stakeholders:

e-Framework Integrity Group (eFIG); end-users.

Audience:

e-Framework Integrity Group. Planned derivative works include materials to appear on the e-Framework web site explaining how service classifications are defined and used in documenting the components of the e-Framework.

Service Classifications

Each of the core components (service usage models, service genres, and service expressions) SHALL be classified in the description using the

classification facets described below. Required and optional classification facets for each component are listed. Values SHALL be provided for each required facet. Values SHOULD be provided for all facets.

Component Service Classification

Each of the components has a set of required and optional service classification facets as listed below. Multiplicities are also shown.

Service Usage Model

- Required Classification Facets: SUM Type, Component Status, Domain, Maturity, Purpose, XOR, Confidence
- Optional Classification Facets: Development Status, Deployment Scale, State Behaviour, Transactional Behaviour, Batch Behaviour, Time Constraint Behaviour, Service End Point, Auth'ed, Protocol Binding

Service Genre

- Required Classification Facets: Component Status, Domain, Maturity, Confidence
- Optional Classification Facets: Deployment Scale

Service Expression

- Required Classification Facets: Component Status, Domain, Development Status, Maturity, Confidence, State Behaviour, Transactional Behaviour, Batch Behaviour, Time Constraint Behaviour, Service End Point, Auth'ed, Protocol Binding, Service Genre Coverage
- Optional Classification Facets: Deployment Scale

The service classification facets are also represented in the matrix.
☑ – required
☐ – optional
☒ – not applicable
(m) – multiple values allowed

Status Facets	**Service Usage Model**	**Service Genre**	**Service Expression**
SUM Type	☑	☒	☒
Component Status*	☑	☑	☑
Domain	☑ (m)	☑ (m)	☑ (m)
Development Status	☐	☒	☑
Deployment Scale	☐	☐	☐
Maturity	☑	☑	☑
Purpose	☑ (m)	☒	☒
XOR	☑	☒	☒
Confidence*	☑	☑	☑

* Values for "Component Status" and "Confidence" will be determined by the e-Framework editors and/or the e-Framework Integrity Group.

Technical Facets	Service Usage Model	Service Genre	Service Expression
State Behaviour	☐	☒	☑
Transactional Behaviour	☐	☒	☑
Batch Behaviour	☐ (m)	☒	☑ (m)
Time Constraint Behaviour	☐	☒	☑
Service End Point	☐	☒	☑
Auth'ed	☐	☒	☑
Protocol Binding	☐ (m)	☒	☑ (m)
Service Genre Coverage	☒	☒	☑

Service Classification Facets

Service classifications are organized into two broad collections:

- *Status Facets*: aspects describing the general status and state of the component. Status facets are applicable to all components.
- *Technical Facets*: technical and behavioural characteristics of the component. Technical facets are primarily applicable to service expressions.

All of the facets, independent of how they are assigned to components, are listed below. Unless otherwise noted, the facets are context free and if a facet is used to classify multiple components, it has the same meaning for each of the components.

NB: Facets described are not listed in any particular order.
NB: A particular facet may not be applicable to all components.
REC: Facets should be presented in a recommended order in service component publications.

Status Service Classification Facets

SUM Type: The kinds of domains in which the SUM is used. Values include:

- Domain: the component is specific and applicable only to a specific set of domains.
- Core: the component commonly recurs either within and/or across domains. For a SUM to be classified as Core it must be approved as such by the eFIG. This value SHALL only be used for service usage models (it implies a core SUM, aka service pattern).

NB: The value "core" for a submission MUST be approved by the eFIG.

Component Status: The overall status of the component and its documentation within the e-Framework. Values include:

- Placeholder: Component has been identified but not documented.
- Unapproved: Component has been documented, but the documentation has not been reviewed or approved. The documentation has been released.
- Approved: Component has been fully documented (for a specific version of the documentation) and the documentation has been released.
- Withdrawn: Component description has been withdrawn.
- Superseded: The component and documentation has been replaced by a newer version or the component has been replaced by a newer component.

> **NB:** While it might be useful to know when a version of a component is replaced by another version, this is a role for the versioning and change log data. Classifications SHALL NOT be used to describe eFIG workflow or relationships between items.

Domain: Focus, or area, of activity in which the component is principally used within an IT system. Areas of activity are generalised from the context of higher education and values include:

- Learning and Teaching: encompasses “e-teaching” and “e-training” as well as “e-learning”
- Administration: sometimes referred to as “e-administration”
- Libraries: includes all the services typically available through library systems
- Research: sometimes referred to as “e-research” and “e-science” but encompasses investigative study in both the sciences and humanities
- IT-Services: describes IT support systems that enable specific IT applications utilised in the other four domains
- Common: The component is common and equally applicable to all five domains above.

The value may be repeated if the component is used within multiple domains.

Development Status: Status of the development of the component. Values include:

- Proposed: Component has been proposed but has not been developed.
- Developmental: Component is being developed and tested.
- Prototype: Component is in prototype deployment.
- Production: Component is in full, production-level deployment.

Deployment Scale: The level of deployment or use within the community. Values include:

- Isolated: Component is deployed in a small number of or limited instances.
- Ubiquitous: Component is widely deployed.

NB: Deployment includes both implementation and use without a deployment, e.g., in a document or model.

Maturity: The measure of maturity of the component. Values include:

- Immature: Component is unreliable and/or rapidly evolving.
- Mature: Component is stable and potentially a legacy component.

NB: A component may be mature but not widely deployed (e.g., legacy). Similarly a component may be widely deployed, but still in development (e.g., beta).

Purpose: The purpose of documenting this component, describing its intent and use by the submitter or the e-Framework. Values include:

- Exemplar: The component has been developed and documented as an exemplar. Use and deployments may not exist.
- Application: The component has been developed and documented to describe an existing application or SERVICE. Production-level use and deployments do exist.
- Modelling: The component has been developed and documented as part of a modelling exercise. Use and deployments may not exist.
- Toolkit: The component has been developed and documented as part of developing a toolkit. Production-level use and deployments of the resulting toolkit(s) may not exist.

NB: Do not confuse the purpose of documenting this component with the purpose of the SERVICE itself.

XOR: Whether the service usage model is defined exclusively in terms of service genres or service expressions. Values include:

- Service Genres: The service usage model is defined in terms of service genres.
- Service Expressions: The service usage model is defined in terms of service expressions.

Confidence: A measure that indicates the level of consensus the e-Framework Partners have in the accuracy of the documented service component. Values include:

- Low
- Medium
- High

Technical Service Classification Facets

Technical facets imply characteristics of the resulting service implementation(s). As such, the facet descriptions are given in terms of the characteristics of the service implementation(s).

State Behaviour: How a service implementation invocation is related to any other service implementation invocation. Values include:

- Stateful: Behaviour of a service implementation invocation is related to another invocation through recording and communications of state information between the invocations.
- Stateless: Individual service implementation invocations are stateless and not correlated with other invocations.

Transactional Behaviour: How a service implementation processes data as transactions. Values include:

- Transactional/ACID: The service implementation processes data as transactions and the transactions SHOULD BE (are designed to be) ACID safe.
- Transactional/Non ACID: The service implementation processes data as transactions and the transactions are not ACID safe.
- Non Transactional: The service implementation has no transactional behaviour.

ACID: Atomic, Consistent, Isolated, Durable
NB: Transactional behaviour implies stateful processing

Batch Behaviour: The type of processing of data or document sets by the service implementation. Values include:

- Individual: The service processes a single data resource per invocation. Results or errors are returned pertaining to the operation or data resource as a whole.
- Batch: The service processes a collection of documents or batch of data per invocation. Results or errors are returned both for the overall operation and for each element in the batch.

Time Constraint Behaviour: How the service implementation deals with time constraints. Values include:

- Hard Real Time: The service implementation must meet hard real time constraints.
- Soft Real Time: The service implementation must meet soft real time constraints.
- None: There are no real time constraints on the service implementation.

Real Time – Correctness of an operation depends on both logical correctness of operation AND the time when the operation is performed.
Hard Real Time – Operation must be completed within a specific time window to be successful (e.g., making on online share purchase).
Soft Real Time – Operation can be completed outside of a specific time window, but with degradation of service (e.g., losing frames in a real time video application).

Service End Point: A description of the role of the component's data/resource processing relationship with other components. Values include:

- Provider: The component is primarily a data/document provider, accessed by other components.
- Requestor: The component is primarily a data/document requestor, accessing other components.
- Transcoder: The component transcodes data (requests and provides).

Auth'ed: A description of how the service implementation incorporates authentication or authorization controls. Values include:

- Auth'ed: The component includes or relies upon internal authentication and/or authorization controls.
- Not Auth'ed: The component does not include or rely upon internal authentication or authorization controls.

REC: IESR has a controlled vocabulary to describe aspect of access control. For the e-Framework such details are deferred to the component service description or can be captured in keywords, not a controlled vocabulary.

Protocol Binding: The type of communications protocol binding for the service implementation. Values include:

- Web Service: The service implementation is implemented using web services standards, e.g., WSDL and SOAP.
- SOAP: The service implementation uses SOAP, but not other web services standards. There is no WSDL definition.
- REST: The service implementation formally follows the REST model.
- HTTP: The service implementation uses only HTTP protocols.
- Other: The service implementation uses some other protocol. (e.g., TCP, UDP)

Service Genre Coverage: Whether the service expression includes all of the features of the parent service genre. Values include:

- Full: The service expression includes all of the features of the parent service genre.
- Extended: The service expression includes all of the features of the parent service genre plus additional features not in the service genre.
- Subset: The service expression includes only a subset of features of the parent service genre.
- Overlapping: The service expression includes a subset of features of the parent service genre plus additional features not in the service genre.