

# **Technical Principles**

## **Enterprise Architecture: Governing Principles**

## Contents

Business Applications and the British Council .....	3
Maximising Microsoft Infrastructure Benefits .....	4
Industry Standards .....	5
Buy not build .....	6
Non-Vendor Specific Solutions .....	7
Security Standards.....	8
Common Data Model .....	9
Data Duplication.....	10
Solution Characteristics .....	11
Systems Management .....	12

### Business Applications and the British Council

<b>Principle</b>	<p>We re-use existing application functionality (including SAP) to support global business requirements where the existing solution meets all mandatory requirements i.e. it is considered "good enough".</p> <p>When this is not possible the British Council will follow industry best practice and select new global business systems based on fit to business requirements and total lifecycle costs.</p>
<b>Rationale</b>	<ol style="list-style-type: none"> <li>1. The British Council can leverage its investment in SAP by utilising existing functionality to support additional business requirements - this can be done with minimal development effort utilising existing British Council staff skills</li> <li>2. The British Council does not have an integration strategy or a middleware implementation to support a portfolio of solutions - this adds significant direct and in-direct costs to any non-SAP global solution and will impede business agility by increasing complexity</li> <li>3. This supports the British Council's drive for application convergence and for a reduction in our platform costs</li> <li>4. The British Council cannot afford "best-of-breed" applications that oftentimes come with a significant cost premium</li> <li>5. The British Council as a Non-departmental Public Body must always be seen to follow UK and EU rules and regulations as well as internal policies</li> <li>6. This approach will ensure that The British Council gets the best value for money for business solutions</li> </ol>
<b>Implications</b>	<ol style="list-style-type: none"> <li>1. The Architecture Group needs a good understanding of the capabilities of existing SAP functionality and the capabilities of additional modules</li> <li>2. The Architecture Group needs a good understanding of the SAP technology roadmap</li> <li>3. To support this strategy the British Council needs to develop an integration architecture to facilitate</li> <li>4. Architecture governance needs to be involved in the decision making process because 'mandatory requirements' is open to interpretation</li> </ol>
<b>Obstacles</b>	<ol style="list-style-type: none"> <li>1. Existing system capacity</li> <li>2. Limited IT capacity to understand existing capabilities - it may just be easier to pick a new system</li> </ol>
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. The functionality of the British Council SAP system needs to be catalogued and communicated effectively in business terms</li> <li>2. This principle needs to be communicated to all senior business managers</li> <li>3. The Architecture Group needs to develop integration architecture</li> <li>4. Architecture governance needs to be put in place</li> </ol>

### Maximising Microsoft Infrastructure Benefits

<b>Principle</b>	Solution designs maximise use of capabilities provided by the Microsoft infrastructure.
<b>Rationale</b>	<p>British Council has invested heavily in its IT infrastructure (including network, platform, collaboration, integration), and capabilities to support that infrastructure.</p> <p>Leveraging the infrastructure capabilities will help to maximise return on IT investment.</p>
<b>Implications</b>	<ol style="list-style-type: none"> <li>1. Infrastructure design must consider requirements across the British Council</li> <li>2. We must ensure that solutions make best use of capabilities provided by the infrastructure</li> </ol>
<b>Obstacles</b>	<ol style="list-style-type: none"> <li>1. Could potentially conflict with following principle (Industry Standards)</li> </ol>
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. Review the infrastructure architecture to ensure that it has considered requirements across the British Council</li> <li>2. Review all solutions to ensure that they make best use of capabilities provided by the infrastructure</li> </ol>

### Industry Standards

<b>Principle</b>	The proposed architecture and constituent technologies must support open, industry standards and avoid proprietary interface protocols and Application Programming Interfaces (API's).
<b>Rationale</b>	<p>The use of industry standard interfaces will reduce development cost, integration costs and the time required to implement new functionality. Where this is not possible all interfaces must be implemented through published API's.</p> <p>A non unified approach will result in the creation of stand-alone technologies with no points of integration.</p>
<b>Implications</b>	<ol style="list-style-type: none"> <li>1. Architecture Group need to create a set of patterns covering approved integration standards</li> <li>2. Systems which do not conform to standards may need to be wrapped to enable integration</li> </ol>
<b>Obstacles</b>	<ol style="list-style-type: none"> <li>1. Compliance of existing systems and services</li> <li>2. Visibility – it may be difficult to establish which existing systems and services are compliant</li> </ol>
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. Create integration standards</li> <li>2. Review existing system to identify which components may need to be wrapped</li> <li>3. Develop roadmap for compliance</li> </ol>

### Buy not build

<b>Principle</b>	New functionality is met using commercial-off-the-shelf (COTS) products rather than developed using bespoke internal solutions.
<b>Rationale</b>	In a mature market, it is generally the case that commercial-off-the-shelf (COTS) products will have a much greater range of functionality than internal built applications. This is because the product is built by a company specializing in the specific domain, producing the functions required both now and in the future for a large range of customers.
<b>Implications</b>	<ol style="list-style-type: none"> <li>1. The total cost of ownership for any system includes not only development and implementation costs, but subsequent maintenance, development and support costs which must be considered</li> <li>2. Self Build needs to be considered in the following circumstances:             <ol style="list-style-type: none"> <li>a. In a new market where commercial products do not currently exist</li> <li>b. In-house development may be necessary to build competitive edge</li> </ol> </li> </ol>
<b>Obstacles</b>	<ol style="list-style-type: none"> <li>1. Lack of skills within the business community to specify requirements at system and service level</li> <li>2. Capacity to do this</li> <li>3. Culture, within some specific areas of the organisation</li> </ol>
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. The Technical Architecture Community should participate in all product selection processes to ensure architectural principles are being adhered to</li> <li>2. Ensure that the business community are aware of the requirements for buy not build and that this is clearly communicated</li> </ol>

### Non-Vendor Specific Solutions

<b>Principle</b>	No application or solution should dictate dependency on components from a single vendor.
<b>Rationale</b>	Inter-dependencies between application, hardware and operating software, database management software etc increase complexity and impede agility, as well as increasing the costs of change.
<b>Implications</b>	<ol style="list-style-type: none"> <li>1. Architecture should be based on open standards</li> <li>2. The architecture must not create interdependencies between layers (no tight-coupling) - the lack of inter-dependencies will allow layers of the architecture to be replaced independently of the other services</li> </ol>
<b>Obstacles</b>	<ol style="list-style-type: none"> <li>1. May be difficult to persuade users to give up their old systems</li> </ol>
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. Ensure that the architecture (enterprise and solution level) is based on open standards</li> <li>2. Ensure that the architecture and solutions does not introduce tight-coupling between components</li> </ol>

### Security Standards

<b>Principle</b>	Security standards must exist for all technologies used in the architecture
<b>Rationale</b>	<p>Security standards are specifications for how a particular technology is to be configured and deployed. All technologies specified by the architecture must be risk assessed to ensure that they can be implemented and operated without introducing unacceptable risk.</p> <p>Examples of security standards include:</p> <ul style="list-style-type: none"> <li>o Operating system hardening (typically one standard per OS version)</li> <li>o MS Exchange configuration</li> <li>o Email client configuration</li> <li>o Remote access tools e.g. CITRIX, Terminal Services, SSH</li> <li>o Database configuration</li> <li>o Web server configuration</li> </ul>
<b>Implications</b>	<ol style="list-style-type: none"> <li>1. Changes in individual technologies must be tracked and the accompanying security standards updated</li> <li>2. Governance processes must include the auditing of systems against the security standards</li> <li>3. Change management processes must include checks against applicable security standards</li> </ol>
<b>Obstacles</b>	<ol style="list-style-type: none"> <li>1. The architecture might require technologies for which British Council security standards do not yet exist</li> </ol>
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. Ensure that the governance process allows for the timely risk assessment of new or changed technologies</li> <li>2. Ensure that the governance process allows for the time limited and risk assessed dispensation of compliance with security standards</li> <li>3. Maintain a list of "approved" and/or "recommended" security-assessed technologies to simplify architectural decisions</li> </ol>

### Common Data Model

<b>Principle</b>	All data stored in systems, and exchanged between components in systems must be mapped to the enterprise wide logical data model.
<b>Rationale</b>	An Enterprise Data Model is an integrated view of the data produced and consumed across an entire organization. This high level model should be mapped to the physical implementation of data across all systems. This standard approach will assist in the identification of duplicate data across the enterprise, allow common naming standards and descriptions to be defined for each data entity and provide the basic building block for the generation of business intelligence.
<b>Implications</b>	<ol style="list-style-type: none"> <li>1. A comprehensive entity-relationship data model will need to be developed and maintained</li> <li>2. Data definitions or metadata covering key data entities need to be produced</li> <li>3. The Enterprise Data Model (EDM) needs to be created</li> </ol>
<b>Obstacles</b>	<ol style="list-style-type: none"> <li>1. Capacity and competency</li> <li>2. Sign-off from the business</li> </ol>
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. Architecture Group to identify a single department and identify, define and describe agreed data entities</li> </ol>

### Data Duplication

<b>Principle</b>	The duplication of data across systems must be avoided.
<b>Rationale</b>	Duplication of data across systems results in increased development and integration effort and complexity to ensure data integrity. Unnecessary duplication may result in the development of additional interfaces or may provide different interpretations of standard reports used within the business.
<b>Implications</b>	<ol style="list-style-type: none"> <li>1. Data ownership within the business must be clearly identified</li> <li>2. Data must be mastered in one place</li> <li>3. Standardised mechanisms must be provided to enable data consumers to access data</li> </ol>
<b>Obstacles</b>	<ol style="list-style-type: none"> <li>1. Capacity and competency</li> <li>2. Culture, may be difficult to persuade current data owners to give up ownership</li> </ol>
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. Agree data ownership</li> <li>2. Evolve Enterprise Data Model ensuring that data is mastered in only one place in a way that delivers incremental business benefits</li> <li>3. Create data access architecture providing standard enterprise-wide data access mechanisms</li> </ol>

### Solution Characteristics

<b>Principle</b>	All application solutions must be designed to provide the levels of service specified by the business (resilience, availability, etc.).
<b>Rationale</b>	Over as well as under engineering of resilience capability can be equally undesirable. In the one instance the degree of resilience may not meet business requirements with potential impact upon the quality of service, while alternatively; over-engineering results in additional unnecessary costs which may never be recovered.
<b>Implications</b>	<ol style="list-style-type: none"> <li>1. Business owners must define expected levels of service for their applications and services from which the overall system design can be created</li> <li>2. Need to define levels of resilience, availability, performance, usability etc.</li> </ol>
<b>Obstacles</b>	<ol style="list-style-type: none"> <li>1. Lack of understanding of criteria to be used when making business decisions</li> </ol>
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. Create guidelines for making business decisions</li> <li>2. Define levels of resilience, availability, performance, usability etc.</li> </ol>

### Systems Management

<b>Principle</b>	All systems must be designed to be managed remotely via electronic interfaces.
<b>Rationale</b>	Systems Management capability will provide the operations and application support groups with information pro-actively which will identify faults and potential areas of concern within the environment. This will allow problems to be identified and fixed more readily and allow pro-active application and system maintenance to be performed before system services become impacted.
<b>Implications</b>	<ol style="list-style-type: none"> <li>1. As a minimum the system should provide:             <ol style="list-style-type: none"> <li>a. Configuration Management of platform and application.</li> <li>b. Fault/alarm Management</li> <li>c. Performance Management</li> </ol> </li> </ol>
<b>Obstacles</b>	<ol style="list-style-type: none"> <li>1. Currently the service management processes required to drive these functions are not fully in place</li> </ol>
<b>Actions</b>	<ol style="list-style-type: none"> <li>1. Ensure that systems management requirements are designed into all solutions (both at the platform and application level)</li> <li>2. Establish the service management processes to utilise remote system management</li> </ol>