|  |  |
| --- | --- |
| Title | Architecture Policy |
| **Owner** |  |
| **Version Number** |  |
| **Document Id** |  |
| **Primary Audience** |  |
| **Document Location** |  |
| **Next Review Date** |  |

# Contents

[A. Objective 1](#_Toc191043896)

[B. Context 1](#_Toc191043897)

[C. Requirements 2](#_Toc191043898)

[D. Solution Architecture Principles 5](#_Toc191043899)

[E. Data Architecture Principles 7](#_Toc191043900)

[F. Application Architecture Policy 10](#_Toc191043901)

[G. Data Architecture Policy 18](#_Toc191043902)

[H. Integration Architecture Policy 22](#_Toc191043903)

[I. Security Architecture Policy 25](#_Toc191043904)

[J. Availability, Resilience and Disaster Recovery Architecture Policy 28](#_Toc191043905)

[K. Infrastructure Technology Policy 30](#_Toc191043906)

[L. Remote Management Architecture Policy 33](#_Toc191043907)

[M. Impact of quality attributes on design decisions 36](#_Toc191043908)

[N. Contact for Queries and Guidance 37](#_Toc191043909)

[O. Supporting & Reference Materials 37](#_Toc191043910)

[P. Review & Approval 37](#_Toc191043911)

[Q. History 38](#_Toc191043912)

# Objective

The core architecture principles provide a guide for architecting a sound solution. A solution should strive to comply with the Succession Group's architecture policies and principles as much as possible. The principles are not mutually exclusive; in certain cases, they may conflict regarding the business requirements and the final implementation. It is therefore not possible to adhere completely to all these principles. A Group Architectural Board will however scrutinize each solution against these principles as a guide to solution quality.

# Context

## Business Context

These policies and principles concern the technical architectural areas of Data Applications, Information, Integration, Security and Technology.

Solution architecture presents the existing AS-IS solution or target estate against the new TO-BE solution, in terms of the application and technology architectures, to illustrate how the new solution will fit in the target estate or replace the existing solution, and what the impact on existing peripheral systems might be. It also demonstrates how the solution complies with the prevailing core architectural principles and specifically points out any necessary deviations from the principles laid out here.

Data architecture ensures smooth digital transformations and new application implementations because data underpins every aspect of our systems. The better the data quality and the better the data connections between system components, the better the system experience and the easier it is to automate processes, reports on business aspects, gain insights predict trends and make decisions. Data Architecture is both an art form and a science for connecting, managing and making available data to end-users to enable automation, self-service reporting, data-led decision-making and creation of new business opportunities.

## Regulatory Context

Succession’s requirements for meeting its obligations under the UK General Data Protection Regulation as well as the FCA’s expectations regarding how it collects, manages, processes, secures and disposes of client data are set out in the respective project’s high-level designs, which are aligned to this Group Architecture Policy.

## Scope

This policy covers the entire life cycle of data and the implementation of infrastructure and application solutions within Succession.

## Risk Appetite Alignment

The nature of IT Solutions risks is such that they can never be fully mitigated. The Boards have no appetite for poorly architected or unsupportable solutions and the resulting business impact that such solutions may have.

# Requirements

## Roles & Responsibilities

**Enterprise Architect**

The Enterprise Architect is responsible for developing and maintaining the organisation’s overall technical architecture vision. The Enterprise Architect’s responsibilities are to:

* Ensure that Succession’s IT systems are aligned with its business objectives and strategy.
* Develop and maintain the architecture roadmap aligned with business strategy.
* Ensure architecture decisions meet both short-term and long-term business goals.
* Oversee the integration of new technologies, ensuring they align with current architecture.
* Act as the liaison between business and IT, communicating architectural vision

**Business Analyst**

The Business Analyst works closely with business units and IT teams to ensure the Enterprise Architecture Vision aligns with business needs. They gather and document business requirements, both functional and non-functional. The Business Analyst’s responsibilities include:

* Analyse business processes and recommend process improvements to enhance efficiency.
* Translate business needs into technical requirements for architects.
* Collaborate with the Product Management Team to ensure product development aligns with overall business and technology strategies.
* Conduct feasibility studies and cost-benefit analyses for new initiatives.

**Solution Architect**

The Solution Architect focuses on designing specific solutions that fit within the enterprise architecture framework. They ensure that these solutions not only meet business requirements but also align with the broader technical architecture and strategy set by the Enterprise Architect. The key responsibilities include:

* Design and implement technical solutions that address business needs and align with the enterprise’s architecture.
* Evaluate new technologies and assess how they fit into the existing architecture.
* Ensure scalability, security, and reliability of solutions while adhering to established architectural guidelines.
* Work closely with domain architects to ensure that individual solutions fit within broader domain strategies.

**Data Architect**

The Data Architect ensures that data is accessible, available, manageable, well understood, secure, private, compliant, and access-governed to enable automation, data-led decision-making and the creation of new business opportunities. The Data Architect acts as an interface between the IT project delivery function and the business stakeholders and works closely with the solution architect to achieve this. The key responsibilities also include:

* Ensure one version of the truth, consistent and unambiguous data across the enterprise.
* Foster connectivity to the data stores. Without ready access and available technical interfaces, the data is meaningless to the business.
* Design a scalable strategy that handles increasing data volumes, data computations and data consumption
* Strive for good data quality that is sufficiently complete for the business's purposes.
* Ensure that data is represented across the enterprise consistently and can be described and categorized using business terminology.
* Implement governing functions based on people-based rules and automated processes, that ensure data quality, compliance, ownership, accountability, cleansing, standardisation, categorisation, security, and privacy.
* Work closely with domain architects to ensure that data approaches fit within broader domain strategies.

**Data Owners**

All data must have an owner responsible for understanding the appropriate risks and implications of the owned data getting breached. The Data Owner is also responsible for assigning the data sensitivity classification of the data and must ensure that it is consistently protected throughout its life cycle up to destruction in a manner commensurate with its sensitivity, regardless of the type of media that the data resides on and what technology is used to handle the data.

**Data Stewards**

All data must have a Data Steward responsible for the data quality, completeness and timely relevance, and who can verify the data sources and technical interfaces to the sources. The Data Steward receives and responds to reports of data errors.

**Architecture Governance Lead**

The Architecture Governance Lead ensures that architecture decisions comply with established policies, guidelines, and standards. This role is vital in enforcing discipline in how architecture is developed and managed, preventing scope creep, and ensuring architectural integrity. The key responsibilities are:

* Define and implement governance structures to ensure architecture compliance across all projects.
* Ensure that the right tools and processes to support the Group’s governance initiative
* Monitor architecture projects to ensure they follow enterprise-wide standards and guidelines.
* Report to the Board on architecture governance, highlighting compliance and identifying areas for improvement.
* Participate in the Change Management Board (CAB) to support a project’s sign-off.
* Convener and manager of the Group Architecture Board (GAB), which ensures that all architecture decisions align with business goals and policies.

## Policy Application

The degree of architectural governance and the artefacts that need to be produced depend on the project’s complexity and business criticality. See the table below for guidance. The complexity metrics 1, 2 and 3 correspond to Light, Medium and High Complexity and will be advised by the GAB:

|  |  |  |  |
| --- | --- | --- | --- |
| **Architectural Artefacts:** | **Complexity:** | | |
|  | 1 | 2 | 3 |
| Functional Requirements | **X** | **X** | **X** |
| Non-Functional Requirements |  | **X** | **X** |
| Initial Solutions Architecture | **X** | **X** | **X** |
| High Level Solution Architecture |  | **X** | **X** |
| Data Architecture |  |  | **X** |
| Group Architecture Board Approval | **X** | **X** | **X** |

## Monitoring, Testing, Reporting, and Measurement of Effectiveness

The effectiveness of the Group Architectural policy will be measured through:

* The creation of architectural artefacts commensurate with the complexity of the projects
* The existence of an infallible support structure for every system on the estate
* The identification and cataloguing of design patterns for reuse when delivering solutions to familiar problems

## Breaches

It is impossible to adhere to all the policies and principles for a design and pragmatic decisions have to be made to meet the business requirements within the allotted time, allowed budget and available technologies. Deviations from these principles and policies must be marked up as Exceptions in the supporting architectural design artefacts, which the Group Architectural Board must then approve.

# Solution Architecture Principles

## Simplicity

IT solutions must be:

* Understandable
* Simple to use
* Simple to deploy
* Simple to operate
* Simple to support

**Rationale**:

Solutions designed in this way will be quicker and more cost-effective for IT to implement, operate, maintain, change and support them. If solutions are easy for users to understand then training will be optimised, and user errors should be minimised. Often, the simpler solution tends to also address the business requirement that it attempts to address better than the complex solution.

**Implications**:

Introducing an IT solution must simplify business operations and fulfil a stated business capability. Use as few components as possible to meet the business's needs. Always look to reduce duplication of capabilities. Architecture designs must describe where complexity is being introduced and what alternatives were considered. Necessary complexity must be hidden from the user and may require additional IT systems and services. Applications must have a clear purpose and well-defined scope. Applications should be designed to have modular functionality to make it easy to modify and reuse. Be especially careful when adding new functionality to existing applications. The user interface must present our employees and customers with application front ends that have a common look and feel. They should be intuitive to use, guiding the user to complete the business process. IT documents must use clear language and avoid the assumption of technical or domain-specific knowledge. IT systems and capabilities must be consistently named.

## Repeatable

IT solutions must be re­usable and support the same business processes across all business departments.

**Rationale:**

To enable business growth, IT solutions must be designed to be re­usable and repeatable wherever possible across the departments, to minimise development, deployment effort and training costs.

**Implications**:

Common processes and systems must be established wherever possible. A business and technical operating model is slowly evolving and the patterns that this establishes should be applied to new business and IT systems. All solutions must be standards-based and open industry standards must be used where possible. Service Oriented Architecture approach: Requirements should be met by building or using existing services and consuming them in applications, rather than by building monolithic applications. Service orientation give future flexibility and allows a better response to new requirements. Service orientation lowers the cost in the long term as it facilitates re­use.

## Agility in Design

IT solutions must be flexible and scalable to enable future business growth and process change.

**Rationale**:

The rapid proliferation of business initiatives results in new IT requirements. IT needs to be able to anticipate and respond to these new requirements quickly and in an agile way to support the business strategy. We therefore need to architect solution designs to be flexible and scalable without requiring architectural change. It is simpler and more cost-effective to build scalability into solutions at the start.

**Implications**:

IT solutions must be designed to be capable of being deployed rapidly in new countries. IT solutions must be capable of being re­used without the need for architectural change. Do not use UK-specific volume estimates when designing new solutions. IT solutions must be capable of being scaled, for example up to larger numbers of users or transactions. Artificial constraints (e.g. number of stores, products, promotions, customers, etc.) must not be built into solutions. IT solutions must be capable of being both vertically and horizontally scaled. IT solutions must be capable of being deployed in different ways, e.g. centralised or distributed deployments to be flexible to the specific needs of different markets.

## Supportable

IT solutions must be capable of being deployed, operated and supported remotely.

**Rationale**:

Well-supported applications provide a reliable automation framework for the business to operate in. Easily supported application reduce the operation and maintenance costs of applications and helps in managing a central support operation.

**Implications**:

Supportability must be built into solution design as standard. All applications, services and infrastructure must be designed with appropriate resilience, be fault tolerant, recoverable, able to be monitored and be able to emit activity alerts to a standard event and monitoring service. Using standard technical infrastructure building blocks to reduce variation in the IT estate simplifies infrastructure support, lowers maintenance costs and increases the overall reliability of infrastructure.

## Cost-effective

IT solutions must be designed with total cost of ownership in mind over the entire life cycle of the solution

**Rationale**:

The investment in IT solutions must consider the overall long-term cost of hosting, supporting, and decommissioning the solution, not just the short-term costs of development and deployment.

**Implications**:

Business needs to drive IT developments, which means that a validated business case must exist before architectural design begins. Before starting any development or purchasing of services, Total Cost of Ownership (TCO) analysis should be carried out for all solutions and fed back to the business after a solution has been designed. Group deals with existing vendors must be leveraged where possible. Check for existing contracts and take them into account, together with the relevant longer-term vendor roadmaps when considering alternative solutions. Architecture designs must always consider the overall IT systems portfolio and include any resulting decommissioning of systems. Leverage existing repeatable technical building blocks to reduce costs. Use one shared and central infrastructure to support multiple systems.

## Secure

IT solutions must be designed to be appropriately secure.

**Rationale**

We must protect our corporate information and assets in appropriate ways to meet regulatory (see [Ref. 1), legal and customer needs. We must protect against unauthorised access and both intentional and accidental modification. UK law and regulations provide strong rules for protecting confidential information. We must be fully compliant with all requirements while looking forward to ensuring future compliance.

**Implications**:

Security requirements must be integrated into the entire architecture process. Architecture designs must reflect policies to minimise improper use of data and security violations. Security must be applied consistently and compliance able to be monitored. Information security controls need to be clearly defined so security and risk are balanced and understood. Auditing requirements must be considered in all architecture designs where appropriate.

# Data Architecture Principles

Data Architecture is both an art form and a science for connecting, managing and democratizing data to enable automation, self-service reporting, data-led decision-making and the creation of new business opportunities.

## Data is a shared asset

**Rationale:**

Enterprises that start with a vision of data as a shared asset ultimately outperform their competition. Instead of allowing departmental data silos to persist, these enterprises ensure that all stakeholders have a complete view of the company, such as improved products and also improved business insights, by being able to correlate data from all systems and business functions. The result of this is better product design, improved corporate efficiency and better business decisions.

**Implications:**

* Data democratization should make the data available to those who need to access it.
* The sharing of data within the organization between persons and processes must be controlled on a need-to-know basis and least-privilege basis, and governance processes must exist for the request, issuance and revocation of access to the data.
* Access and understanding of data make for better business decisions
* Inaccessible data has no inherent value.

## Provide the right interfaces for users to consume the data.

**Rationale:**

It is not realistic to put all data in one place to achieve the vision of data-driven products. For systems to benefit from a shared data asset, one needs to provide controlled yet usable interfaces that make it easy for systems and business users to consume that data.

**Implications:**

The interfaces can be in the form of an OLAP interface for business intelligence, an SQL interface for data analysts or a real-time ReST-full API for systems. It is essential that the technical interfaces to the data are available to allow the systems to access required in a system-native way, and for business users to use the tools they know and are right for the job that they need to perform.

## Ensure security and access controls.

**Rationale:**

The emergence of unified data platforms like Snowflake, Google Big Query, Amazon Redshift, and map-reduce offerings (Hadoop et al) has necessitated the enforcement of data policies and access controls directly on the raw data, instead of in a web of downstream data stores and applications.

**Implications:**

The emergence of new data security projects necessitates a unified security-oriented approach, so consider using technologies that allow you to architect for security, and deliver broad self-service access, without compromising control.

## Establish and maintain a common vocabulary.

**Rationale:**

By investing in an enterprise-wide data approach, such as an enterprise data hub, enterprises can now create a shared data asset for multiple consumers across a complex product supported by many vendors, or the business with its many departments. The arising ambiguity of the meaning of the data needs to be disambiguated using an established common vocabulary. With a shared vocabulary, less time will be spent disputing, disambiguating or reconciling results and more time driving improved product output and business performance.

**Implications:**

Data users must analyse and understand the purpose and content of the data in terms of a common vocabulary. System design, product catalogues, data warehouse dimensions, provider hierarchies and KPI definitions all need to be aligned to this common vocabulary, regardless of how users consume or analyse the data.

## Data must be trustworthy

Reliable data is accurate, authentic, has integrity, is usable, and can confidently be used for decision-making, analysis, and other purposes. Trusted data is critical for making informed decisions.

**Rationale:**

Systems and end-users can have a frustrating experience if data is not cleansed, incomplete or misaligned, or stored in unsafe environments where it can be tampered with. This reduces the perceived and realized value of the system’s underlying data.

**Implications:**

By investing in processes that curate data, be it manual or automated, the Succession Group improves its ability to realize the value of its shared data assets. Typical data problems that can be mediated by such an approach are:

* Inaccurate data
* Outdated data
* Duplicate data
* Incomplete data
* Inconsistent data
* Irrelevant data
* Uncatalogued data
* Unstructured data
* Non-compliant to regulatory standards
* Data of unverified provenance

## Eliminate data copies and movement.

**Rationale:**

Every time data is moved there is an impact on one or more of the following:

* Cost
* Accuracy
* Time

**Implications:**

The fewer times data must be moved, the better. Part of the promise of cloud data platforms and distributed file systems is a multi-structure, multi-workload environment for parallel processing of massive data sets. These data platforms scale linearly as workloads and data volumes grow. By eliminating the need for additional data movement, modern enterprise data architectures can reduce cost (time, effort, accuracy) and optimize overall enterprise data agility.

# Application Architecture Policy

## Pattern-based applications

Applications must be built according to defined patterns. Where no patterns exist because the application is a new type of application (business or technology-wise), design patterns should be extracted and documented from the successful implementation.

**Rationale:**

Standardised applications developed by reusing existing patterns will reduce development times and also make support simpler and reduce costs.

**Implications:**

Existing patterns must be used when custom application development is done.

**Refer to Principles:**

* Repeatable
* Simple

## Service orientation

Applications must reuse existing services. If an appropriate service does not exist, a new service must be developed which can be reused in the future. Self-contained applications must not be developed where services could be used instead.

**Rationale:**

The reuse of services for multiple applications should enable flexibility and responsiveness and save development time and cost.

**Implications:**

Where they already exist, appropriate existing services must be reused. Where a service does not already exist, a new service must be developed rather than building a self-contained application.

**Refer to Principles:**

* Repeatable
* Cost-effective
* Simple

## Consistent, intuitive user interface

Applications must follow the appropriate Succession Group standards and guidelines for user interface design.

**Rationale:**

Our users should have a consistent experience across Succession Group applications, with a common Succession Group look and feel that is intuitive to use and simple to support.

**Implications:**

All internally developed applications must follow the appropriate standards and guidelines for the user interface. Single sign-on must be provided where possible. Customer-facing applications must always follow the Succession Group brand guidelines.

**Refer to Principles:**

* Simple
* Supportable

## Help

All applications must provide the user access to an appropriate level of help and support, guiding them to complete the business process. Application development must seek to make the user interface as simple as possible.

**Rationale:**

Our applications must facilitate user productivity and be intuitive to use. This policy reduces training overheads and support costs; and helps to minimise errors due to system input issues.

**Implications:**

Applications must be intuitive to use and include appropriate user help and guidance via interactive help and/or the use of prompts.

**Refer to Principles:**

* Simple
* Supportable

## Repeatable

Applications must be architected to be repeatable.

**Rationale:**

Applications must be architected to be repeatable across the Succession Group so that we can develop once and deploy many times. It is most Cost-effective to build the ability to use applications internationally when they are initially designed.

**Implications:**

Applications must be designed for repeatability, internationally across the Succession Group and should consider character set; multiple languages; multiple currencies and compliance requirements. Where global or regional instances are required, applications must not require separate instances to support multiple languages, currencies, legal requirements, etc.

**Refer to Principles:**

* Repeatable
* Cost-effective

## Application remote support

Applications must be designed to be supported remotely.

**Rationale:**

To ensure that we have IT that works every day; by standardising our support, costs will be re-used.

**Implications:**

All applications must be designed to be supported remotely. All support elements of an application (e.g. error logs) must be in English, even if the user interface is in a local language.

**Refer to Principles:**

* Supportable

## Application operation framework

Applications must follow Group standards for systems operations and support framework. This entails code development, release management, monitoring, asset management and configuration control.

**Rationale:**

To ensure that we have IT that works every day.

**Implications:**

All applications must follow the release management, systems operations and support standards.

**Refer to Principles:**

* Supportable

## Application scalability

Applications must be scalable.

**Rationale:**

Applications must be able to support future business growth.

**Implications:**

All applications must be designed to be able to be horizontally (scale-out, e.g. more servers) and/or vertically scalable (scale-up, e.g. more memory, more CPUs). Application design must not introduce artificial constraints, e.g. number of banks, innovations, products, promotions, or customers.

**Refer to Principles:**

* Agile
* Repeatable
* Cost-effective

## Application design and coding

The application design and coding standards must be followed when designing and building applications.

**Rationale:**

Applications must be built in a standard, reusable way, following our internal standards.

**Implications:**

All applications must follow our design standards. All applications must follow our coding standards.

**Refer to Principles:**

* Simple
* Repeatable

## Application configurability

Applications must be designed to be configurable and must not use hard-coded, fixed values and settings.

**Rationale:**

Applications must be developed so that they can easily be reused.

**Implications:**

Applications must be designed to be configurable and to use parameters. Hard-coded fixed values and settings must not be used in applications.

**Refer to Principles:**

* Agile
* Repeatable

## Industry standards-based

Applications must use the Succession Group's preferred relevant industry standards.

**Rationale:**

Using open industry standards will enable easier engagement with our partners and help us and our partners to reduce costs.

**Implications:**

All applications must adopt the relevant open industry standards.

**Refer to Principles:**

* Simple
* Agile
* Cost-effective
* Repeatable

## Application investment

Application investment and development must be aligned to the appropriate enterprise architecture roadmap. Decisions on application investment must take the results of the application investment review and criteria into account.

**Rationale:**

Application investment must be aligned with our IT strategy and roadmaps.

**Implications:**

Our systems portfolio must be actively managed. Applications must not be developed beyond their useful life. All obsolete applications must be decommissioned. Regular investment reviews of all applications must be undertaken to identify potential applications for decommissioning.

**Refer to Principles:**

* Cost-effective

## Business process metrics capture

Applications must capture business process metrics to enable continuous improvement and optimisation.

**Rationale:**

Ensures that we capture the metrics that are required in order to monitor and improve our business processes. It is more Cost-effective to build this capability into applications when they are designed.

**Implications:**

Applications must capture business process metrics and make them available for reporting where applicable. For example, capturing how long a process/task takes to complete on a device.

**Refer to Principles:**

* Cost-effective

## Testable

All applications must be testable and follow the testing standards.

**Rationale:**

Applications developed must meet requirements; facilitates quick, simple and Cost-effective regression testing.

**Implications:**

It must be possible to configure the application to run in a test environment using a test database without change to the code.

**Refer to Principles:**

* Supportable
* Cost-effective

## Resilience

Applications must be resilient and recoverable.

**Rationale:**

This policy aids the operation, maintenance and support of applications to reduce overall cost.

**Implications:**

Application resilience must be designed to be proportionate to risk. Interface validation, error handling and recovery mechanisms must be built into the design of all applications using standard patterns. Application level resilience approach example: Bad data must not bring down the application but must instead be hospitalised by the application.

**Refer to Principles:**

* Supportable
* Cost-effective

## Business continuity

Business continuity and disaster recovery must be built into applications according to the required service level.

**Rationale:**

Provides appropriate business continuity in the event of failure

**Implications:**

Business continuity and disaster recovery, including failover, must be considered in the application design according to the defined service level.

**Refer to Principles:**

* Supportable
* Cost-effective

## Packaged application implementation

When implementing a packaged application from an external vendor customisation must be avoided. We must fit our business processes with the chosen application package rather than change the application. Applications must be hosted internally as a first choice.

**Rationale:**

To help ensure simple, cost-effective applications. Vanilla packages are easier and less costly to maintain and upgrade than packages that have been customised with Succession-specific developments. Packages are primarily implemented to support processes where Succession Group has little competitive advantage and there is therefore unlikely to be significant business benefit in customising them. It is usually more cost-effective to host applications internally.

**Implications:**

If Succession Group's processes do not fit the processes in the package, then we must first try to change our processes to fit the package, before customisation is considered. Package customisation must be kept to a minimum and must be signed-off prior to development. Approved package customisation must be delivered as a bolt-on to the application, with as little change to delivered code as possible to ensure that upgrades can be delivered effectively. Suppliers must sign-up to ongoing improvement and management of customisations. Third party hosted applications must be avoided where possible; when external hosting is agreed the solution must be capable of being monitored and have clear SLAs.

**Refer to Principles:**

* Repeatable
* Agile
* Cost-effective
* Supportable

## Application deployment

Applications must be packaged to enable them to be easily deployed.

**Rationale:**

Reduced maintenance and support of multiple versions. Code is quicker and easier to deploy.

**Implications:**

Application code is configuration-managed and version-controlled. Code is managed in one place; code changes are bundled into releases. It must be easy to identify customisations or localisations. It must also be possible to revert to a previous version. An application must have a common code base and every physical implementation (e.g. each implementation in individual local countries) must have the same level of patching.

**Refer to Principles:**

* Supportable
* Cost-effective

## Green

Applications must be able to cope with being powered down as part of power saving.

**Rationale:**

In order to achieve green IT initiatives, applications must be capable of being powered down as part of power saving.

**Implications:**

Power saving measures must be considered in application architecture. Where possible it must be possible to power down applications as part of energy saving measures when they are not in use.

**Refer to Principles:**

* Cost-effective

## Domain Names

DNS names must be used to locate resources and devices on the network.

**Rationale:**

Isolates applications from underlying network and IP addressing which makes it easier to make infrastructure changes.

**Implications:**

Applications use names rather than hard coded IP addresses.

**Refer to Principles:**

* Simple
* Information

# Data Architecture Policy

## Master data

There must be only one system that masters a given data collection.

**Rationale:**

For enterprise data to be managed effectively, there can only be one primary source for each data element. Otherwise, inconsistent and erroneous data may result. It is less costly to maintain timely, accurate data in a single source, and then share it, than it is to maintain duplicative data in multiple sources. Multiple repositories for referential information add complexity and increase support and maintenance. Proliferation of multiple data stores would create disconnection in the enterprise and increase the complexity and costs associated with enterprise-wide integration.

**Implications:**

Better integration of data. Wherever possible data must be consolidated into one place where it can be maintained and used as the master data source. This ensures that the data is available to both central and local applications and users. This excludes redundant data being held for archive, backup and disaster recovery purposes.

**Refer to Principles:**

* Simple
* Repeatable

## Data model

All data models including those produced for local solutions must consider both regional and group dimensions.

**Rationale:**

Promotes consistent, re-useable and effective information systems. Promotes better understanding of the data. Data is defined consistently throughout the enterprise to enable data sharing, and the definitions are understandable and available to all users.

**Implications:**

Data models must be developed to support future regional and international re-use. Concepts such as as multi-currency, regionalisation and multi-language should all be considered.

**Refer to Principles:**

* Repeatable
* Simple

## Data Sensitivity Classification and Protection

All data held on Succession Group-managed systems must be classified according to Succession Group's Data Classification Policy. See [Ref. 2]. This determines how the data is stored, transferred, managed and protected. Data must be protected according to it the sensitivity of its content.

**Rationale:**

Protecting data costs money. By understanding the data classification and assessing the impact of a data breach, an appropriate security approach can be selected to protect the data.

**Implications:**

Solution architecture must document all data repositories and data flows. This must include a business function of the data, what fields are present, volumetrics, sensitivity, type of data, in-transit encryption, data ownership and Data Classification.

## Succession Group data model

Corporate data must be described in a Succession Group way.

**Rationale:**

Standardisation reduces the duplication of data entry and maintenance efforts. Interoperability of systems and data allows business processes to be more flexible and adaptive to changing needs. Poor information does not become a barrier to business growth.

**Implications:**

A Succession Group corporate data model must be maintained to ensure a common understanding of data. The Succession Group corporate data model describes common data definitions and terminology and must be used to describe data in a consistent way across applications. There is a cost to standardisation and a trade-off analysis should be done.

**Refer to Principles:**

* Agile
* Repeatable
* Simple

## Common data access toolsets

Strategic tools are provided for data access and must be used for maintaining and accessing databases.

**Rationale:**

Re-use of common toolsets to maintain and access data means that the cost of tool licenses are reduced and skills and support can be focused on specific recommended technologies.

**Implications:**

We have standard tools that are fully understood within the business and supportable centrally by the Succession Group Technology Team. Standard tools must be used by new projects that require data access.

**Refer to Principles:**

* Supportable

## Persisting data

Data storage must be classified with retention and recoverability requirements.

**Rationale:**

Data owners must be accountable for the effective and efficient management of data. The accuracy, currency and security of data are management concerns best handled by data owners.

**Implications:**

Data quality must be maintained for each data attribute at the business logic layer. Data scrubbing and cleansing processes must be maintained to improve the quality of the data.

**Refer to Principles:**

* Supportable

## Data quality

Data quality is the responsibility of the master system.

**Rationale:**

Data and its quality must be maintained in the master system.

**Implications:**

Applications must have sufficient data verification and validation at the user interface to ensure data quality. Data cleansing must be carried out in the master system to ensure the quality of data is maintained.

**Refer to Principles:**

* Simple
* Repeatable
* Cost-effective

## Data lifecycle and purging

The lifecycle of data must be managed. Data that no longer has any value to Succession Group or external legal entities must be purged.

**Rationale:**

Data must be managed throughout its lifecycle including so that the data maintained in our systems is up to date and relevant. If data is no longer of any value to Succession Group it is using up valuable storage resource for no benefit. This data must be purged to prevent it interfering with useful data by increasing search times. We are obliged to provide any data we hold for legal reasons; as such we should purge it if it is not required.

**Implications:**

Data lifecycle management must be included in solution architectures. There must be a process in place for purging data that is no longer required. The process for identifying what data needs to be purged, how to design for ease of purging (i.e. database partitioning), how data will be purged and how often needs to be defined as part of a solution architecture.

**Refer to Principles:**

* Cost-effective

## Data Archiving

An archive solution/tier must be provided for the long-term storage of data.

**Rationale:**

An archive solution will automatically move data to the lowest speed and cost tier.

**Implications:**

Solution architectures must include the provision of an appropriate archiving solution for the long term storage of data. Applications must have an appropriate archive solution or tier for long term data storage.

**Refer to Principles:**

* Cost-effective
* Agile

## Data replication

Creating additional copies of data outside the master source must be avoided.

**Rationale:**

Data replication costs time and money. Storing the same data in different locations increases storage resource and can lead to multiple versions of the truth.

**Implications:**

Solutions must not be designed which replicate data; solutions must always use the one master source of data that already exists. Services on the master source of data are the best solution and must be used where possible. If replication is unavoidable, data synchronisation must be considered in the solution architecture and design and applications must not become a secondary source of this data. Databases must not be copied to third party systems.

**Refer to Principles:**

* Cost-effective

## Data security

Data must be appropriately secured and audited to comply with corporate security policies and legal requirements.

**Rationale:**

To minimise improper use or loss of data, which could have serious business and legal consequences.

**Implications:**

Data must be appropriately locked down so that only people who need to access to the data have authorisation. Data access control must be considered in the architecture. There must be a complete audit trail of the data from source system to end user: Auditing and reconciliation of data must be performed at various stages throughout the data lifecycle. Data should not be secured to the point that responding to open records requests becomes prohibitively expensive.

**Refer to Principles:**

* Secure

## Data Keying

Surrogate keys must be used where possible when a new database is created that uses data from an existing database.

**Rationale:**

Using surrogate keys rather than natural keys avoids closely coupling analytical data to operational systems. This avoids dependencies being built against natural keys in the source system.

**Implications:**

Use surrogate keys to map to the natural keys in the source system.

**Refer to Principles:**

* Agile
* Supportable

# Integration Architecture Policy

## Integration layer for integration

The integration layer is intended for integration only, not to make up for the idiosyncrasies within applications.

**Rationale:**

The integration layer must only be used for integration to keep it simple and to optimise performance.

**Implications:**

Complex business logic must not be implemented in the integration layer; and new data must not be generated as part of the integration layer. If required such logic must be encapsulated within a service that the integration layer consumes. Any activity other than content based routing or non-semantic transformation should be encapsulated in a discrete service.

**Refer to Principles:**

* Simple
* Repeatable
* Cost-effective

## Integration layer re-usability

The integration layer must be re-usable across group. Systems must be designed to maximise the re-use of existing interfaces in the integration layer where possible.

**Rationale:**

The flexible and open exchange of data between applications will minimise complexity and application spaghetti. Re-using the integration across the group will make solutions repeatable and increase Cost-effectiveness.

**Implications:**

The integration layer must be designed to be re-used across the Succession Group. Solutions must re-use existing interfaces in the integration layer where possible. Common coding standards must be followed for ease of maintenance. Ensure interfaces are versioned to ensure backward compatibility.

**Refer to Principles:**

* Repeatable
* Cost-effective

## Succession Group common form

Data in the integration layer must be translated into a generic Succession Group specific format, commonly referred to as a common or canonical form

**Rationale:**

Translation of all data in the integration layer to the Succession Group canonical form will help to remove direct coupling between systems.

**Implications:**

All Succession Group data which flows through the integration layer must be translated to Succession Group common form.

**Refer to Principles:**

* Agile
* Repeatable

## Event driven data integration

Event driven data integration must be considered before batch integration. If possible, synchronisation should occur through the use of an event driven architecture in preference to batch transfers.

**Rationale:**

Integration must not add delay to a business process. Frequency of synchronisation must be appropriate to business need.

**Implications:**

All data integration must be achieved through an event driven architecture where possible. For batch processes to be used for data integration there must be an approved business reason.

**Refer to Principles:**

* Simple
* Agile
* Supportable
* Cost-effective

## Push not pull

Data must be pushed once it has been created rather than polled for and pulled by the target system.

**Rationale:**

The source system will know when the file has been created and as such it is more efficient to send the data once it has been created rather than to keep checking to see if it is there. In addition data must be extracted to file rather than being pulled directly from the source database by the receiving system.

**Implications:**

When creating a schedule for extracting data consider the next step to be the transmission of the data.

**Refer to Principles:**

* Simple
* Repeatable

## Chunking

Large files (>100Mb) must be split into smaller chunks.

**Rationale:**

Transmission of large files forces a sequential processing. Chunking a file into smaller pieces means that there is less stress on the network. Fewer issues with restart provision and allows the receiving application to begin processing more quickly.

**Implications:**

It is important to understand the potential volume of data that will be transmitted and to design accordingly.

**Refer to Principles:**

Supportable

## Schema validation

All XML documents sent through the integration layer must be validated at the point of entry against an agreed schema definition (XSD). The same applies for JSON, YAML

**Rationale:**

The passing of invalid messages can result in problems within the integration layer.

**Implications:**

When designing messages formalise the design in an XML schema definition.

**Refer to Principles:**

* Cost-effective
* Supportable

# Security Architecture Policy

## Solution Security

All solutions must comply with legal and regulatory requirements and internal good practice that supports the requirements that originate from the data classification policies.

**Rationale:**

We must run the business safely for ourselves and our customers and be in compliance where required (although there are no obvious examples at Succession Group as yet), in order to protect the Succession Group brand, shareholders and the customers.

**Implications:**

All solutions must show om the architecture documentation detail how this policy is met. Any architectural exceptions must go through the appropriate governance and exception processes.

## Defence in Depth

It is risky to rely on a single security control on a system to enforce security on sensitive data or system access. By using multiple layers of security such that failure of one layer does not compromise the entire solution.

**Rationale:**

Security solutions can fail and when this happens it is important that our IT is not exposed as a result. If one security control fails there must be at least another one that will protect data and system from compromise. If an attacker has to break multiple security layers to gain access to our systems and information then they are less likely to succeed.

**Implications:**

The solution architecture must use multiple layers of security where appropriate, and this needs to be documented and agreed to be either be sufficient or not, or excessive.

## Reference architecture, patterns and standards

The solution architecture must re-use existing security infrastructure and patterns and comply with standards and support the requirements that arise from the data classification policy. Where no appropriate existing standard exists, a new standard must be defined that includes the necessary security protections and is reviewed for compliance with information security policies.

**Rationale:**

Re-use can shorten implementation times, reduce the technical and user-journey complexity, reduce initial and ongoing costs, and deliver consistent, high levels of security and minimal customer disruption to new solutions with minimal effort. Tactical solutions are by their nature not secure or Cost-effective and weakens Succession Group's overall security position.

**Implications:**

Solution architectures must re-use existing appropriate security infrastructure (e.g. vulnerability scanning, security incident and event management, web application firewalls, intrusion detection, etc.) and patterns. Where patterns and standards do not exist, new ones must be defined and approved to ensure future compliance. Non-compliance must be clearly identified in solution architecture documentation and agreed with by the Enterprise Architect.

## Know our risks

Solution architectures must be risk-assessed against the threats, vulnerabilities and impacts that the system is perceived to have. The impact of losing service availability, integrity or confidentiality of data must be understood. This assessment must be continued once a solution is operational to continually evaluate and mitigate against new threats and vulnerabilities.

**Rationale:**

We need to take a risk-based approach to understand emerging security threats and the risk of potential impact to the Succession Group. This will maintain the risk profile to a level acceptable to Succession Group.

**Implications:**

Solution architectures must be risk-assessed against all possible threats, weaknesses and impacts. We must understand how a breach of confidentiality, integrity, or availability can happen and what the impact would be.

## Data accessibility drives segregation model

Data with similar security or accessibility requirements must share the same security zones. Segregation models must be respected, e.g. network segregation and tiers. All network zones must be classified by trust according to the trust standard and controls must exist at these boundaries to provide segregation and reduce the identified risks.

**Rationale:**

This policy reduces the number of security solutions required but maintains the correct level of security for each set of data. Segregation models are core to regulatory compliance schemes such as PCI-DSS. They also prevent inappropriate access to systems and data. Failure to appropriately segregate networks, environments or solutions may lead to inappropriate access to customer data, weaken security and lower customer confidence in Succession Group. Our inability to comply with regulations such as PCI-DSS and the Data Protection Act will lead to fines and reputational damage.

**Implications:**

Solutions must be architected to ensure that data with similar security or accessibility requirements share the same security zones. All network zones must be classified by trust (trusted, semi-trusted, un-trusted, restricted) according to the trust standard and controls must exist at these boundaries to provide segregation and reduce the identified risks.

## Access authentication & authorisation

Authentication must be done against the Group's directory solution. Authorisation must be done against groups in the Group's directory solution.

**Rationale:**

Authentication and authorisation is the cornerstone of effective security. Using a single strategic solution builds upon existing processes to deliver the simplest, lowest cost and best solution for staff and customers. It is simpler to manage a single directory rather than multiple authorisation soultions. The use of alternative authentication or authorisation solutions would result in a more complex environment where it would be inherently more difficult to determine if access rights had been appropriately granted. In turn this would result in higher costs for audit and regulatory compliance checks.

**Implications:**

Solutions must use the Group's directory solution for authentication and authorisation. For each and every component in the solution, the solution architecture must describe how authentication and authorisation of users will be performed for that component. Sensitive authentication data must be protected; a solution must show clearly where this risk is identified and how the risk is to be mitigated against. Any non- compliances should be identified and go through the appropriate governance process. In situations where an exception is agreed, additional architecture documentation must detail how the authentication and authorisation will work and how the solution will be secure.

## Access control - least privilege

Access control must be implemented with least privilege and separation of duties. All access permissions must be defined in terms of business and IT roles. Role definitions and information must be mastered by a role authority and new roles must go through a review and sign off process. Users/roles must be given the minimum permissions necessary to perform their role.

**Rationale:**

Excessive access or multiple accesses to related systems can introduce opportunities for fraud and unauthorised access to go undetected. Roles based access control reduces the administrative burden required to manage system access rights. By grouping access rights into roles and assigning users to the roles required for their job, we can be assured that users have been granted only the minimum privileges required to perform their job. Failure to use roles-based access control will increase the cost and complexity required to administer users and access rights. There is a strong possibility that users may be granted inappropriate access to systems, resulting in unnecessary risks being placed upon the organisation.

**Implications:**

Users and support staff must only have the level of access to systems, data and networks that is necessary for their role. Every solution must demonstrate that access to it is understood and is being controlled. Where certain kinds of access lead to greater risks, these risks must be identified and authentication methods orescribed that mitigate them. For example, management access to a solution can be exploited. Where management access is required, a solution must identify the risks and demonstrate how these will be protected against. User entitlement reviews must be carried out on a regular basis. Any non-compliances should be identified and go through the appropriate governance process. In situations where an exception is agreed, additional architecture documentation must detail how the authentication and authorisation will work and how the solution will be secure.

# Availability, Resilience and Disaster Recovery Architecture Policy

Solutions must meet business requirements for availability and resilience and must be recoverable. Business continuity and disaster recovery must be built into solutions according to the required service level.

**Rationale:**

If we are under attack our customers must not be prevented from accessing our services. If our solutions fail in a disaster scenario we must continue to provide service to customers with little or no obvious impact. If solutions do not implement robust failover/disaster recovery then it is highly likely that customers will not continue to transact with us, ultimately impacting our financial position and industry standing.

**Implications:**

Business continuity and disaster recovery, including failover, must be considered in the solution design according to the defined service level to meet business requirements. Resilience must be proportionate to risk. The solution architecture must fully document the failover of each component in the solution, describing (i) how component failure is detected, (ii) how traffic is rerouted (iii) how the solution protects against loss of service to customers and (iv) how the service can be recovered in a disaster scenario.

## Security and vulnerability management

Threats to systems and associated vulnerabilities must be continuously assessed and reviewed. Solutions must therefore be capable of being continuously monitored and must produce security logs.

**Rationale:**

We must have the capability to identify security weaknesses in our systems in order to understand and prevent attacks.

**Implications:**

The solution architecture must document how each solution will be capable of being continuously monitored. For example, the architecture must show how the solution will be capable of having patches applied and anti-virus signatures updated.

## Log and audit

Security relevant events must be logged and an audit trail kept.

**Rationale:**

The ability to log and audit the actions of all parties and processes which interact with our solutions is a key component in ensuring that our security approach remains effective. Failure to log and/or audit events may result in attackers going unnoticed, malicious access going undetected and ultimately loss of customer or commercially sensitive data.

**Implications:**

The solution architecture must document how each solution component performs logging and auditing; this must include describing those events beyond the standard patterns which should be logged.

## Security control automation

Automated security controls must be used in preference to manual security controls wherever possible in order to provide consistent security.

**Rationale:**

Manual processes are vulnerable to error. Automated solutions achieve consistent performance and will make IT security cheaper and simpler to operate.

**Implications:**

Security controls must be automated wherever possible. Manual security controls must only be used by exception. Our systems must be capable of being continually monitored for weaknesses and potential threats.

## Third-party access

All third-party access must be identified and appropriate controls applied to manage the risk.

**Rationale:**

Giving third parties access to our network and/or systems is a risk. We must therefore identify additional controls, for example, by limiting scope and privileges of a third party’s access.

**Implications:**

All access by third parties must be identified and must be well monitored, authenticated and limited as far as possible; using appropriate controls to manage the risk.

# Infrastructure Technology Policy

## Shared Technical Services (Infrastructure Blocks)

Infrastructure must be built as a series of shared technical services using established industry patterns backed up by group-wide commercial deals. Solutions must use these shared technical services.

**Rationale:**

Enables efficient, repeatable use of infrastructure. Speeds up deployment of infrastructure and simplifies support by reducing variation and duplication. Group deals ensure that we get the best value for money.

**Implications:**

Architecture, design, delivery, support capability, SLAs and Group deals need to be in place before a technical service can be rolled out. Funded via central core IT budget or usage-based model. Typically, these services are cloud-based.

**Refer to Principles:**

* Repeatable
* Supportable
* Secure
* Cost-effective

## Scalable and Flexible Infrastructure

Shared technical services must be able to scale to accommodate organic growth and new projects. We use virtualisation to flexibly (and dynamically) allocate resources to workloads.

**Rationale:**

We can grow and reduce infrastructure on demand without having to replace components. Virtualisation technologies allow physical infrastructure to be shared and provide logical separation of resources that can be allocated (in a controlled way) to guarantee application performance.

**Implications:**

We deploy infrastructure that can scale without requiring downtime. We have the organisation, processes and tools in place to manage dynamic virtualised environments in an automated way.

**Refer to Principles:**

* Agile
* Repeatable

## Standard Infrastructure Components

Standard infrastructure components must be used to support solutions. A shared technical service is based on one design and is comprised of standard infrastructure components.

**Rationale:**

Reduces variation in the IT estate which makes support simpler for Succession Group, lowers maintenance costs and increases reliability. Accommodating change will be easier

**Implications:**

We may have to trade off increased functionality against the need to maintain IT across a large estate

**Refer to Principles:**

* Simple
* Supportable

## Disaster Recovery (DR)

Critical systems must have disaster recovery in place. Recovery times in the event of a disaster must be agreed and signed off by the business. The recovery order (priority) for systems in the event of DR must be established and agreed with business owners.

**Rationale:**

Systems that are critical to keep Succession Group trading must be available in the event of a disaster. Provides clear prioritisation to enable effective targeting of resources after an incident. Allows an effective DR plan to be created.

**Implications:**

Critical systems are identified, system DR is fully integrated into business continuity plans. DR plan, site, infrastructure and procedures are in place.

**Refer to Principles:**

* Supportable

## Centralised Infrastructure

Infrastructure must be centralised into a data centre wherever possible. Workloads that cannot be run in a data centre must be run on local shared infrastructure.

**Rationale:**

Using centralised infrastructure makes more efficient use of resources and is simpler to operate. If we have to deploy a workload locally we must ensure the infrastructure can be used by other applications / workloads

**Implications:**

We need to ensure that the business SLA can be met and that the impact on network bandwidth and cost is understood. Governance and accountability for managing local shared infrastructure (e.g. Succession Group server in a bank) needs to be clear

**Refer to Principles:**

* Cost-effective
* Supportable
* Simple

## Infrastructure investment

Infrastructure investment and development must be aligned to the appropriate enterprise architecture roadmap. Decisions on infrastructure investment must take the results of the infrastructure investment review and criteria into account.

**Rationale:**

This policy ensures that infrastructure investment is aligned with our IT strategy and roadmaps.

**Implications:**

Our systems portfolio must be actively managed. Infrastructure must not be developed beyond it's useful life. All obsolete infrastructure must be decommissioned. Regular investment reviews of all infrastructure must be undertaken in order to identify potential infrastructure for decommissioning.

**Refer to Principles:**

* Cost-effective

## Industry standards based

Technical services must be based on open standards.

**Rationale:**

Using open industry standards will enable easier interoperability and integration between infrastructure supplied by different vendors and help us to reduce costs.

**Implications:**

All technical services must adopt the relevant open industry standards.

**Refer to Principles:**

* Cost-effective
* Supportable
* Simple

## Development and Test Infrastructure

We must minimise the amount of infrastructure required to support development and test workloads and consider how it can be re-used for DR purposes.

**Rationale:**

It is more Cost-effective to use spare capacity in shared infrastructure (including DR environments).

**Implications:**

Changes to development and test networks will come under change control process.

**Refer to Principles:**

* Cost-effective

# Remote Management Architecture Policy

An infrastructure component must have remote log-in, backup, build and patch management capability.

**Rationale:**

Ensures IT can control and maintain devices remotely and removes the need for dedicated support desktops. Infrastructure can be operated in an automated and centralised way.

**Implications:**

Remote administration is secured and access logged. Service manager, support team, processes and tools are in place.

**Refer to Principles:**

* Supportable
* Secure

## Remote Monitoring

A technical service/infrastructure component must provide an interface to allow the end-to-end health of the service/component to be monitored via centralised support tools.

**Rationale:**

Ensures IT can monitor the health of infrastructure and compliance to SLAs

**Implications:**

Standard monitoring tools are used wherever possible, gaps in monitoring capability are identified before deployment and addressed

**Refer to Principles:**

* Supportable

## Backup

Production data must be backed up. Data backed up for the purpose of disaster recovery must be stored at a different location from the source data.

**Rationale:**

All data should be regularly backed up as agreed with the data owner. A backup on the same site as the source data could be destroyed by the same event.

**Implications:**

We deploy a backup & recovery service with the appropriate processes and resources to test regularly. Data is stored in a different location from the production data.

**Refer to Principles:**

* Secure
* Network Demarcation

## Geographical network logical separation

**Rationale:**

Creates a natural break where routing and security policies can be applied. Instability in one network will not propagate to another.

**Implications:**

Route re-distribution is used to ensure seamless connectivity.

**Refer to Principles:**

* Repeatable
* Supportable
* Secure

## Network Communication

All devices, shared infrastructure services and applications must use Group standard (IPv4) for network communication. All devices, shared infrastructure services and applications must be capable of supporting IPv6.

**Rationale:**

Worldwide standard protocol for networking. Ensures network inter-operability. TCP is preferred instead of UDP because of the flow control and error recovery capability. Future proofs infrastructure and applications.

**Implications:**

We have phased out use of legacy network protocols such as IPX and SNA

**Refer to Principles:**

* Simple
* Agile

## Group IP Addressing

The Group IP addressing scheme must be used.

**Rationale:**

Simplifies routing and removes duplication of IP addresses.

**Implications:**

All IT systems migrate to the Group IP scheme

**Refer to Principles:**

* Simple

## Network and Authentication Services

Network and authentication services such as Identity, Directory, Authentication, DNS, DHCP, NTP and IP address management must be integrated across Group.

**Rationale:**

Network and authentication services must be integrated to allow a service to be referenced from anywhere within the Succession Group (e.g. global applications).

**Implications:**

We adhere to Group standards for network and authentication services

**Refer to Principles:**

* Simple
* Agile

## Business Unit Separation for Identity Access Management

Separate business units/countries must be in their own Active Directory Forest.

**Rationale:**

Keeping business units/countries separate maintains security and makes it easier to divest the business in the future.

**Implications:**

We have a standard design to enable centralised support of identity and access management and provide authentication for Group applications. We have a standard way of federating multiple directories to provide a global directory.

**Refer to Principles:**

* Secure
* Agile

## Minimise Environmental Impact

We must reduce the overall environmental impact of IT equipment (e.g., reduce energy usage/packaging, use recycled materials). We must also understand the power usage and carbon emissions of technology and the wider environmental impact of each solution.

**Rationale**:

By understanding the environmental impact of our IT solutions, we can control and reduce it.

**Implications**:

IT solutions must minimise their impact on the environment throughout their lifecycle. Power management and instrumentation must be built into IT solutions and devices.

**Refer to Principles:**

* Cost-effective

# Impact of quality attributes on design decisions

In summary, here is a list of attributes that make for a good-quality design:

**Features**

* The ability of the software feature set to meet the requirements of the project from the customer.

**Extensibility**

* Extensibility measures the ability to extend a system, and the level of effort required to implement the extension. Extensions can be through the addition of new functionality or modification of existing functionality. The principle provides for enhancements without impairing existing system functions.

**Interoperability**

* Interoperability is a characteristic of a product or system, whose interfaces are completely understood, to work with other products or systems, at present or in the future, in either implementation or access, without any restrictions.

**Performance**

* An indicator of how well a software system or component meets its requirements for timeliness. Timeliness is measured in terms of response time or throughput.

**Reliability**

* Software reliability can be defined as the probability of failure-free operation of a computer program in a specified environment for a specified time.

**Durability**

* The software solution’s ability to be serviceable and to meet users' needs for a relatively long time. Software durability is important for user satisfaction. For a piece of software’s security to be durable, it must allow organisations to adjust the software to business needs as necessary.

**Serviceability**

* The ability of technical support personnel to install, configure, and monitor computer products, identify exceptions or faults, debug or isolate faults to root cause analysis, and provide hardware or software maintenance in pursuit of solving a problem and restoring the product into service.

**Conformance**

* Whether a process, product, or service complies with the requirements of a specification, technical standard, contract, or regulation

**Aesthetics**

* These are principles that define a design's pleasing qualities: In visual terms, aesthetics includes factors such as balance, colour, movement, pattern, scale, shape and visual weight.

**Usability**

* Usability is the degree to which software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.

# Contact for Queries and Guidance

Queries are to be initially addressed to the Group Architecture Board.

|  |  |  |  |
| --- | --- | --- | --- |
| Supporting & Reference Materials | | | |
| **Ref** | **Title** | **Document Id** |
| 1 | Information Security Policy |  |
| 2 | Data Sensitivity Classification Policy |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Review & Approval | | |
| **Reviewed by** | **Job Title/Role** | **Date** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Approved by** | **Job Title/Role** | **Approval Date** |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| History | | | |
| **Version #** | **Author** | **Comments** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |