Appendix 1

Applications Management Strategy 2018-23

City of London Corporation

City of London Police

DRAFT

Contents

1. Introduction

This document outlines a proposed strategy for the City of London Corporation ('CoL') and the City of London Police ('CoLP') to drive value-for-money from their ongoing investment in software applications and related services. It is an output of the recent Applications Optimisation Pathfinder project, as is the accompanying 'Applications Register', which provides an inventory of the applications estate.

A good applications management strategy is typically informed by and acts as a bridge between business strategy, the wider IT strategy and the strategies of application vendors. Both CoL and CoLP have published strategic, overarching local plans that are shaping business strategy. Jointly, they have developed a high-level IT Strategy that links to four contributing strategies: Technology Transformation (the infrastructure modernisation that underpins all the other strategies), Applications Optimisation, Information Management and Digital Enterprise.

CoL have already implemented their Technology Transformation Strategy, while CoLP will start this soon. Therefore, it is now appropriate to agree the Applications Management Strategy.

Please note: some key concepts and abbreviations used in this document might not be familiar to general readers and therefore have been explained in a Glossary at the end.

2. Purpose of This Strategy

This strategy is intended to guide decision-making by business units, the corporate centre, the IT Division and their technology/service partners in relation to managing the lifecycles – i.e. acquisition, ownership and disposal – of applications, both individually and collectively.

Managing applications effectively and efficiently is important for the following reasons:

- Applications are where staff and customers interact with information and processes to create business value, with everything else in the IT estate being enablers for this
- A significant amount of money is spent on the procurement and ongoing operation of applications; therefore, efficiencies and savings are desirable and likely feasible
- Enhancement or replacement of key applications will be essential to support the strategic plans of CoL and CoLP, especially proposed new (digital) ways of working
- Business continuity and regulatory compliance are dependent on the proper working of applications, and as a result, applications can be key sources of business risk

Adoption of this strategy is expected to lead to the following beneficial business outcomes:

- Better use of more readily available intelligence about applications to inform decisions on their acquisition, ownership and disposal
- Better management of risk and resources associated with ownership of applications through more structured and proactive operational monitoring and change planning
- Reduction in the size and/or complexity of the applications estate, thus reducing the associated like-for-like cost of ownership and the attack surface for cyber threats
- Reduction in duplication and/or silos of functionality and data across applications, thus reducing wasted effort and easing collaboration across business units

Even after implementation of this strategy, some residual risk and complexity will continue to exist in relation to the applications estate, but at the very least adopting this strategy will provide a way to understand the reasons why and how best to cope.

3. Scope of This Strategy

This strategy focuses on the management of software applications. All pieces of software that <u>are not</u> part of the operating systems for computing infrastructure (laptop/desktop PCs, mobile devices, networks, servers, storage, virtualisation services) are deemed applications.

The applications estate is made up of the following three types of application:

- **System** – business-specific, process-oriented application that contains predefined business-process rules, e.g. Corelogic's *Mosaic* for adults' and children's social care case management, and Northgate's *Paris* for income collection and management
- **Tool** – non-business-specific, activity-oriented application that contains predefined activity-constraining rules, e.g. Microsoft's *Word* and *Excel* for producing documents and spreadsheets, and ESRI's *ArcGIS* for manipulating digital maps
- **Platform** – non-business-specific, function-oriented application that has few rules but allows them to be added, e.g. Microsoft's *SharePoint* and *Flow* for managing content and workflows, and Firmstep's *Forms* for enabling and managing online self-service

The applications estate is also split into two tiers based on how widely each application is deployed. Tier 1 applications are those deployed to all users, while all other applications are deemed to be Tier 2. The tier that an application falls into has implications for how it is supported and maintained, with knock-on impacts for its users.

This strategy covers the management of all these types and tiers of application using an application lifecycle management model. The model identifies a sequence of lifecycle stages and activities together with the roles responsible for undertaking those activities.

This strategy <u>does not</u> cover strategic product, technology and partnership decisions, e.g. recommending the next financial management system or evaluating *blockchain* as a new application model or determining the future of Microsoft as a core technology partner.

4. Strategic Opportunity and Challenge

The application estates of CoL and CoLP are large, complex and varied, with some overlap between the two estates. This creates an opportunity to increase value-for-money through consolidation, simplification and standardisation across multiple areas, but also makes such changes more challenging to achieve.

The changes can best be made by sharing, re-using, extending and integrating applications. Platforms are most suited to this as they meet functional requirements common to many business units, are widely available and are highly configurable. Tools are the next most suited – although they are activity-specific, they can be used by any person or any business unit that undertakes that activity. Systems tend to be least suited as they are often so highly-tailored to the processes and terminology of specific business units that other units whose businesses are essentially similar will still struggle to use them.

Most applications in the estate are systems or tools. Platforms are far fewer and their potential for delivering business benefits is largely unexploited. Some vendors combine their applications into suites, e.g. Microsoft's *Office 365*, which includes *Word*, *Excel*, *SharePoint* and *Flow* amongst other component applications. If most of a suite's applications can be used beneficially, especially with some helpful integration between the applications, then this is likely to offer better value-for-money than similar applications that stand alone.

A key challenge to implementing an applications management strategy is capturing and maintaining the information needed to support it. CoL has historically devolved management of support contracts and vendor relationships for applications to the business units that use them, which makes comprehensive information capture and maintenance more difficult. CoLP has taken a more centralised approach, which makes this easier. An obligation on all stakeholders to share and maintain applications management information will be mandated by policy, helping to maximise its usefulness to applications-related decision-making.

So far, 327 applications have been identified as being part of the CoL estate based on information gathered during the Technology Transformation programme and validated by the Applications Optimisation Pathfinder project. A further 125 applications have been identified in the CoLP estate but not validated – a validated CoLP list will emerge from a separate CoLP technology stack evaluation project currently underway. It is likely that some existing applications have not been identified yet – e.g. those only using the web browser on a PC – as they do not have component software to be installed on PCs, which was the identification method used for Technology Transformation.

This initial list will be subject to change. Some of applications will be retired soon if agreed by their owners. Meanwhile, new applications are being introduced – 14 in the last six months – some of which might have been avoided if this strategy had been in effect.

5. Applications Estate Management Principles

The principles for managing the applications estate have been derived from the general IT principles set out in the previously agreed IT Strategy (identified in **bold** in the list below). The general IT principles have been expanded and adapted to suit the applications context.

Buy not build

A. Source applications in the following order of preference:
    1. Re-use an existing application system or tool
    2. Configure an existing application platform
    3. Upgrade an existing application
    4. Buy a new application
    5. Build a new application

B. Prefer hosting in the cloud to hosting on-premises, in the following order of preference
    1. Software-as-a-Service (SaaS)
    2. Platform-as-a-Service (PaaS)
    3. Infrastructure-as-a-Service (IaaS)
    4. On-premises hosting

C. When selecting a vendor for a new application or service, prefer functional/strategic fit with business and IT over familiarity/popularity of the vendor with the business or IT

Use fewer systems more effectively

D. Avoid data/functionality silos or duplication in applications – properly source an application once, appropriately re-use it often
E. Integrate core applications to create novel solutions or replace niche solutions
F. Upskill users to make the most of solutions

Secure and compliant IT systems and services to support the organisation

G. Maintain up-to-date and accurate information needed for application lifecycle management and application audits, e.g.
    - Support contract and vendor roadmap
    - License count and user log
    - Access control matrix and user permissions
    - Transaction and backup logs
H. Ensure applications are current (version N-1 where possible) and secure (compliant with GDPR, PSN, PCI, etc. as applicable)
I. Keep applications functioning acceptably via maintenance and support aligned to application tier, criticality and vendor contract

Move from complexity to commodity

J. Follow common (open) standards, including for accessibility
K. Design solutions and services to be intuitive, sustainable and repeatable
L. Work with users to optimise/standardise their user experience

When selecting applications for best fit, those with the features listed below (where relevant) should be preferred to those without. Some of these features reflect best practice in support of the principles listed above, others support the new ways of working being promoted in the separate *Digital Strategy*.

i. Enables and encourages self-service
ii. Supports accessibility for people with disabilities
iii. Supports mobile working via a mobile client or responsive design
iv. Provides service-oriented architecture (SOA), application programming interface (API), data model, etc. for integration
v. Supports single sign-on compatible with CoL/CoLP identity and access management (IAM) platform
vi. Encrypts sensitive data in transit and at rest
vii. Logs and reports events to understand application usage and performance
viii. Supports the relational database platforms preferred by CoL and CoLP
ix. Supports multiple separate user groups while maintaining appropriate data separation, confidentiality and integrity
x. Supports use of virtualised computing, networking and storage facilities

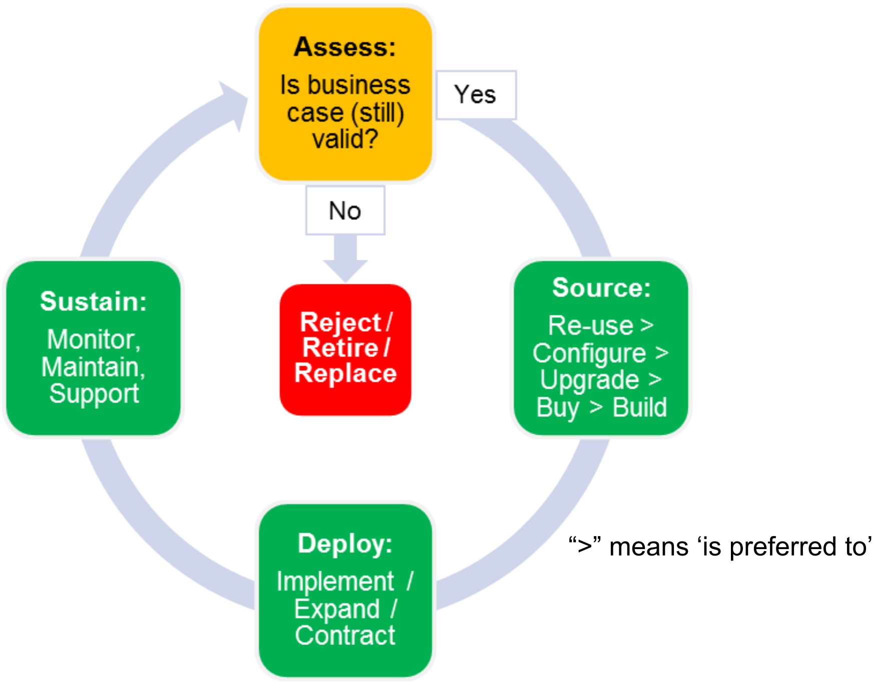
While these principles and features set the standards to which all applications should aspire, this does not mean that all applications not meeting those standards should be immediately retired or replaced. Instead, what to do about such an application should be considered as part of its managed lifecycle, and if some change is deemed necessary, then a plan should be made for carrying out that change at a suitable time. Please see the next section for more on application lifecycle management.

6. Application Lifecycle Management Model

Management of the applications estate collectively is closely related to the model for lifecycle management of each application individually.

An application has a lifespan within CoL or CoLP, consisting of a sequence of phases from acquisition (the beginning of its life), through ownership (the main part of its life), to disposal (the end of its life). To encourage regular reassessment of the application's business case and fitness-for-purpose, ownership can be further broken down into a series of sub-phases.

Within each phase, a cycle of activity is executed that assesses the business case for an application-based solution, and, if the case is valid, then sources, deploys and sustains the application that best matches the case. When the case is no longer valid, the disposal phase is triggered, during which the application is retired or replaced. This is summarised in the diagram below.



The sourcing activity determines how best to obtain an application that matches the business case in terms of fitness-for-purpose, value-for-money and strategic compliance. This activity reflects the preferences set out in Principle A from Section 5, which covers the sourcing of applications. Note that in each ownership phase, 're-use' can mean continuing to use the application sourced in the previous phase (perhaps also as an instance of re-use).

The deployment activity encompasses the implementation of a new or upgraded application, as well as expanding or contracting the user base of an existing application where re-use and/or consolidation are being pursued.

The sustenance activity corresponds to business-as-usual, i.e. the everyday work required to keep the application running as intended and delivering the services that its users expect. Monitoring and maintenance are used to minimise unexpected outages, while support resolves those fewer outages that still occur.

A key output of this model is a roadmap of key events/activities in the lifespan of each application, which together produce an overall schedule for the whole applications estate as well as calendars for the IT Division – especially the Applications team and the Business Partners team – and for each business unit and department. These schedules and calendars are essential for effective work and resource planning and risk management.

Many of the data flows and activity steps required to operate this model are expected to be automated in the future, thereby increasing execution reliability and reducing staff effort.

7. Application Lifecycle Management Policies

The application lifecycle management policies listed below have been agreed by the IT Division senior management team for submission to the IT Sub-committee for approval on behalf of all business units.

1. Each application must have a named Business Owner accountable for its business evolution and use (see Appendix A for more information on this role)
2. Each application must have a named Technical Owner accountable for its technology evolution and technical KPIs
3. All requests for new software must be made using the approved form and in line with the agreed procedures
4. Multiple applications enabling a similar business process, task or functionality must not exist across the estate, including multiple instances of the same application unless signed off by the IT Technical Design Authority
5. All applications must have a valid supplier support and maintenance contract in place
6. Applications must be licensed to meet, but not exceed, the usage requirement (except where bulk buys are more cost effective).
7. All applications must be on the approved Definitive Media Library (DML). Applications discovered on the estate but not in the approved DML will be removed
8. Access to applications must be restricted on a least required privilege approach
9. Applications must use shared infrastructure, servers and instances where appropriate
10. All applications that form systems of record must be backed up in line with the specific application's departmental policy
11. Every application must have a periodic time window agreed with the Business Owner when it may be shut down for maintenance, with notice being given ahead of each shutdown but no further approval sought
12. Only approved tools can be used for the management and monitoring of the environment
13. Applications releases must be signed off by the application's Business Owner
14. The Configuration Management Data Base (CMDB) is the master data repository for all application configuration information (Applications Register in the interim)

15. All design, configuration and operations documentation must be kept up to date and stored in a single location referenced from the CMDB
16. Software Asset Information will be held in SNOW and SCCM
17. All application configurable items must follow the standard naming convention and/or an agreed standard taxonomy where applicable
18. Any application not used for 180 days will be retired from the IT estate unless supported by a Business Case to the IT Division
19. Applications required for evaluation purposes must be requested as a change with a valid project code and agreed retirement date
20. Applications must be hosted on infrastructure suitable to meet, but not by design exceed, the relevant contractual and SLA targets
21. All applications supporting a legislative business requirement, or being a system of record for financial transactions over £100k over a financial year, must be reviewed frequently to ensure they meet legislative, business, and technology changes
22. All information relevant to application lifecycle management must be made available to the IT Division, who will share it with others as appropriate; such information includes but is not limited to each application's business case, support contract and licensing terms, costs and technology dependencies
23. Each application in the estate must undergo a regular review of its Business Value, Technical Quality & Cost by the IT Division in conjunction with the Business Owner

8. Application Lifecycle Management Responsibilities

Implementing the applications lifecycle management policies requires business and IT staff to be assigned the necessary roles and responsibilities.

Please see Appendix A for explanations of the Business Owner and Business Contact roles.

Those responsible for undertaking each activity are also responsible for updating the Applications Register and any other relevant records and documentation to reflect changes arising from the activity.

Note that the current Service Delivery Partner is Agilisys.

Activity	**Responsible**	**Accountable**	**Consulted**	**Informed**
Assess				
• Trap new application request	Service Desk	Service Delivery Manager		Business Partner
• Formulate business case	Business Partner	Business Owner	Business Contact; Applications Team; Enterprise Architect	
• Assess and approve business case allowing sourcing to begin	PMO weekly meeting	PMO	Business Partner	
Source				
• Assess whether existing application can be re-	Applications Team	PMO	Business Partner;	

Activity	Responsible	Accountable	Consulted	Informed
used, configured or upgraded to fulfil the business case			Enterprise Architect	
• Implement reuse, configuration or upgrade of existing application if appropriate	Project Manager; Applications Team	PMO	Business Partner; Enterprise Architect	
• Assess market options for a new application if needed to fulfil the business case	Business Partner	PMO	Procurement; Enterprise Architect; Applications Team	
• Procure new applications if appropriate	Project Manager; Procurement	PMO	Enterprise Architect; Applications Team	
• Assess build options for a new application if needed to fulfil the business case	Applications Team	PMO	Enterprise Architect	
• Execute build of new application	Project Manager; Applications Team	PMO	Enterprise Architect	
Deploy				
• Manage major deployment, e.g. new application or upgrade with significant change	Project Manager	PMO	Applications Team	
• Manage minor deployment, e.g. re-use of applications as-is or an upgrade with little change	Applications Team	PMO		
• Packaging of end-user software components	Packaging Team	[Project Manager or Applications Team]		
• Deployment readiness sign-off	Business Contact; CAB	Business Owner	[Project Manager or Applications Team]	PMO
• Deployment of new software into estate	Service Delivery Partner	[Project Manager or Applications Team]	Business Contact	Application Users; PMO; CAB
Sustain				
• Application operational monitoring	Service Delivery Partner	Applications Team	Business Contact	
• Application patching	Service Delivery Partner	Applications Team	Business Contact; CAB	

Activity	Responsible	Accountable	Consulted	Informed
• Application support	1st Level: Service Delivery Partner 2nd Level: Applications Team	Applications Team		

9. Strategy Implementation Challenges

This strategy has been designed to comply with the existing IT, human resources, finance and procurement principles and policies of CoL and CoLP. Nevertheless, it proposes more proactive, structured and consistent ways of owning and servicing applications than has been the case in the past. This requires some changes to operating practices, roles and responsibilities. Therefore, in adopting this strategy, CoL and CoLP are committing to implementing the changes set out below.

Change	Key Reasons	Accountability
The working practices of Applications team members and Business Partners are revised to reflect the new model while staying within the scope of their existing job descriptions	• Being proactive needs more time, which will be freed up by being more efficient elsewhere • Collaborative delivery needs more co-ordination, which requires staff to take on more responsibility where appropriate	• Head of Applications • Head of Engagement and Change
Training for Applications team in new monitoring and change implementation tools	• New tools need new skills, both to operate and to envisage new benefit opportunities	• Head of Applications
The outputs of the Service Delivery Partner (Agilisys) are revised to reflect the new model while staying within the scope of their existing contract	• SDP must engage more closely with the IT Division and Application Business Owners/Contacts • SDP must be more proactive in application management	• Deputy IT Director (Delivery)

There are inevitably challenges to implementing these changes, key amongst these being the following:

- Perceived limits on contractual obligations of Service Delivery Partner
- Contractual challenges with application suppliers
- Change effort within IT Division and business units
- Staff capacity, e.g. to maintain information, to meet regularly, to be an Application Business Owner
- Staff capability, e.g. to monitor applications proactively, to negotiate sourcing, to configure platforms
- Funding for change, including training and tools for the IT Division and one-off costs to business units for switching from existing applications to those deemed most appropriate for their business need

A separate target operating model lays out these changed ways of working in more detail.

Appendix A: Application Business Owner and Contact Roles

A key role in application lifecycle management is that of the Business Owner. This is the person accountable for realising the value-for-money that is intrinsic to an application's business case. The Business Owner influences this over the application's lifespan by making key decisions about its usage and evolution and the related spend and risk.

A typical application business case envisages achieving value-for-money through the application being used by the appropriate people at the appropriate times and in the appropriate ways, to complete the appropriate tasks and produce the appropriate outputs and outcomes, at the appropriate cost and risk.

Application Business Owner

The Business Owner role encompasses the following accountabilities ('[A]') and responsibilities ('[R]'), supported by the relevant specialists within the business, IT and other services and partners:

- Defining the business case for an application-based solution to address a business weakness, threat or opportunity [A]
- Agreeing the application-based solution recommended after appropriate consideration of the options [R]
- Agreeing the application's business criticality, which in turn defines its security, hosting and support provisions [R]
- Periodic review of the business case for continuing to use the application [R]
- Approving retirement or replacement of the application at the end of its useful life [R]
- Assigning a budget where needed to cover application costs [R]
- Accepting the residual risks associated with the application [R]
- Ensuring the confidentiality, integrity and availability of the information contained within the application [A]
- Approving operational deployment of the application and any of its subsequent upgrades [A]
- Approving the addition, modification and removal of users for the application [A]
- Training and/or guidance of users in the proper use of the application [A]
- Supplier contract performance monitoring and enforcement [A]
- Ensuring that audit recommendations relating to the application are addressed [A]
- Ensuring that the agreed application risk mitigations are put into practice [A]

Application Business Contact

The Business Contact role reports to the Business Owner and is expected to be filled by a person who has operational understanding of the business processes that the application will support and can provide advice and make decisions in relation to usage and change on a day-to-day basis. Business Owner accountabilities will typically translate into Business Contact responsibilities.

Appendix B: Application Criticality, Resilience and Support

Application lifecycle management centres on keeping an application cost-effectively useable for as long as its users need it to be. Applications are typically a bundle of functions and associated data made available to one or more business units in an organisation and perhaps directly to the internal and external customers of those units.

Some application functions and data might be critical to a business unit and/or its customers all the time every day. Other functions and data might be critical at key points in the day, week, month or year. And yet others are not time critical, as long as they are available within a reasonable time window. However, what is critical to one or more business units might not be critical to the organisation as a whole.

An application can fail in totality or one or more of its functions can fail. Some or all of its data can be lost to storage failure or database corruption. In all cases, some sort of restorative action might be needed from IT support providers. Given limited resources, it is necessary for business units to agree with the IT Division ahead of time, application by application, which failures or data losses are to be addressed through preventive measures and/or after-the-event responses, and how, when, by who and at what effort and cost.

Two criticality factors are used to determine the failure/error response/prevention parameters:

- **Recovery Time Objective** (RTO) – where an application (function) has failed, how quickly must it be restored to a usable state?
- **Recovery Point Objective** (RPO) – where up-to-date application data have been lost, a data set from how far back in time is deemed usable?

RPO Instant	C1	C1	C1	C1
RPO Short	C2	C2	C2	C1
RPO Medium	C3	C3	C2	C1
RPO Long	C3	C3	C2	C1
	RTO Long	RTO Medium	RTO Short	RTO Instant

Together, RTO and RPO determine application criticality via the matrix shown at right (business-agreed definitions of *Instant*, *Short*, *Medium* and *Long* are pending), where C1 is highest criticality and C3 is lowest criticality.

Application criticality is used to determine the appropriate level of infrastructure resilience for the application: Gold (the highest level) for C1 applications, Silver for C2, and Bronze for C3.

Resilience is a measure of how well the infrastructure avoids single points of failure and automatically recovers in the event of component failures. The higher the resilience level, the more costly the server, network and storage components and maintenance and support services needed to provide it. Components and services will be shared across applications where possible, in which case only a portion of these costs will be charged to each business unit owning an application that runs on the resilient infrastructure. If dedicated components or services are deemed necessary, then it is likely that the whole of their costs will be charged to the business unit.

In the face of this, a business unit might wish to settle for lower RTO and RPO targets, and hence lower criticality, lower resilience and lower cost. This is their choice, but, unless they overestimated the criticality in the first place, it does increase the business risk. However, it is not acceptable to pay for only Bronze (standard) resilience infrastructure but still expect high criticality support responses/outcomes in the event of application failure or data loss.

Appendix C: Applications Register

The Applications Optimisation Pathfinder project has created an 'Applications Register' as an inventory of all the applications across CoL and CoLP, with information added for each application that can be used to make decisions and plan activities in relation to application lifecycle and estate management. The Register is potentially a stepping stone to an eventual comprehensive IT Configuration Management Database.

The Application Register has been populated by taking the applications list produced by the Technology Transformation programme, validating it and adding further information held by the IT Division on each application. Due to this limited number of sources, the Register is not yet fully populated, but this is expected to be rectified as the application management processes outlined in this strategy are put into practice.

The applications in CoL and CoLP come in many shapes and sizes, reflecting business need and technology availability at the time that they were introduced as well as the evolution in these factors since then. The Application Register tracks these business and technology factors for each application to help identify application strengths, weaknesses, opportunities and threats leading to timely decisions and actions aimed at optimising value-for-money at the estate level.

The business factors are the business needs translated into functional and non-functional requirements. Application functionality falls into 7 broad categories:

- Asset management: recording attributes and tracking status of assets ranging from files to books to buildings to people
- Content management: storing and serving content, e.g. documents or videos, with appropriate access controls and update tracking
- Task management: guiding and tracking tasks – defined as activities with little scope for variation of inputs and outputs – both individually and in groups
- Case management: guiding and tracking cases – defined as activities with a lot of scope for variation in inputs and outcomes – which may be short-term or long-term
- Financial management: maintaining and reporting financial accounts following financial accounting rules
- Relationship management: tracking interactions and outcomes of ongoing relationships with people
- Technical processing: specialist data/content processing, e.g. geographical mapping, statistical analysis or sound and video creation

Applications typically offer functionality from more than one category, but one or two categories can usually be identified as being core to an application's purpose. Many applications duplicate specific elements of functionality, which might be considered a waste. However, this might be justified where a sufficiently different and beneficial combination of functionalities is offered. Business factors include the service and support contract end date.

The technology factors reflect the technologies, services and deployments used to make each application function. Applications range from those that run on one PC to those that run on servers. Some applications require their own software components to be installed on each end-user's PC or other computing device, others can function via the device's built-in web browser software interacting with a web-based application server. Application servers can be hosted in CoL or CoLP premises or in vendor premises or in a third-party environment. Different applications can store their data in different ways, from shared databases to local files. Any of these components might be virtualised, i.e. run on a shared server with other applications while behaving as if running standalone. Technology factors include the end-of-support date for the application version as defined by the vendor.

The Applications Register – an asset management application with software as the asset and capturing all the above factors as asset attributes – is available as a shared facility via SharePoint for reference by business units, the corporate centre and the IT Division.

Note that the Applications Register is separate from and serves a different purpose to the Microsoft System Centre Application Catalogue – the Catalogue lists the applications that can be installed by end-users on demand (subject to approval), which is currently a small sub-set of all the applications recorded in the Register.

Glossary

API (Application Programming Interface)	A part of an application's software that allows other software to interact with it directly in an automated fashion. Using the methods provided by the API, a programmer can have the other software make requests and pass instructions to the application as needed. The application will then respond in the ways defined for the methods and thus understood by the other software. APIs are typically provided by the vendor as an extra.
Cloud Service	Any computing service provided via shared technology such that the end-user only pays per unit of service used. The per unit cost reflects the mixture of variable and fixed costs involved in keeping the service available, in the same way that the per unit cost for using electricity reflects the variable fuel costs and fixed powerplant, transmission and personnel costs involved in its production and delivery. Cloud services may be public (shared with other organisations) or private (shared within the same organisation). Public cloud services are hosted off-site; private cloud services can be hosted either offsite or onsite. The service provider is responsible for maintenance – often invisibly to the user – and support.
Data Model	A description of what data are stored and processed by an application, including the data's format, value restrictions and, often, access rules. This can be used to move data into and out of the application, and can help with information management, security and audits.
GDPR (General Data Protection Regulation)	Data protection good practice rules given the force of UK law. The rules apply to all organisations processing personally identifiable data relating to living individuals, who may be members of the public or employees. Breach of the rules can result in significant fines being imposed by the UK Information Commissioner's Office, as well as reputational damage to the organisation.
IaaS (Infrastructure-as-a-Service)	A computing infrastructure, typically consisting of computer servers, data storage facilities and interconnecting networks and services, that is provided following the cloud service model (see above). The user can choose to run any software on the infrastructure that the infrastructure can support, e.g. platforms and applications (see below), subject to the provider's service rules.
IAM (Identity and Access Management)	The processes and tools required to manage secure user access to IT environments. Ideally – for the sake of simplicity – everyone should have a single personal user identity through which he/she can be authorised to access any element of the IT environment, including applications. In practice, people have many user identities, with associated login details to remember, as many IT environment elements, especially applications, have not yet signed up to centralised IAM.
PaaS (Platform-as-a-Service)	A computing platform – typically meaning software and infrastructure that combine to offer particular functionality without being tied to specific business processes – that is provided following the cloud service model (see above). An example is a database platform, where database functionality is provided but what is stored in the database and the business rules attached to that are not determined by the provider.
PCI DSS (Payment Card Industry Data Security Standard)	A set of standards for securing personal and financial data relating to the use of payment cards, which has been agreed by the payment card industry. Organisations wishing to process card payments must comply with these standards, although some aspects may be devolved to a suitably compliant processing partner. If there is a compliance breach, the breaching organisation may be barred from processing payments until the cause of the breach has been fixed and can face financial penalties.

PSN (Public Services Network)	A UK government network for connecting all sectors and agencies of government, including local authorities. There is a stringent code of connection designed to ensure security of data and systems across the network. Failure to comply with the code can result in an organisation being cut off from the network, meaning that it might no longer have timely access to data and systems vital to its business.
SaaS (Software-as-a-Service)	A software application, and its underpinning platform and infrastructure, provided following the cloud service model (see above). The end-user still has to provide an end-user computing device to access the application.
SDP (Service Delivery Partner)	An external provider of services for hosting, managing, monitoring, maintaining and/or supporting applications, working closely with the IT Division and with business unit customers.
SOA (Service-Oriented Architecture)	Principles and patterns of application design where key functions of the application are exposed as programmatically accessible services for use by other software. SOA serves much the same purpose as an API (see above). However, SOA comes from a service-oriented approach from the start of application development that recognises the application as being one part of an interconnected business/software ecosystem, whereas APIs tend to be added at the end of application development as a bonus.