ICT Project Guidance

System Security

Author:

Sky Sigal, Solution Architect

## Description

This document describes key security controls expected to be applied to systems before they are given permission to be deployed to a Production Data environment for interaction by end users.

## Synopsis

The document summarises reasonable baseline security controls to consider and verify implementation to cover risks relevant to delivery Stakeholders, Providers, Networks, Environments, Devices, Systems Architecture, System Design, Auditing, Sessions, Authentication, Roles and Permissions.

## Contents

[Description 1](#_Toc145261324)

[Synopsis 1](#_Toc145261325)

[Contents 2](#_Toc145261326)

[Introduction 4](#_Toc145261327)

[Background 4](#_Toc145261328)

[Outcomes 4](#_Toc145261329)

[People 5](#_Toc145261330)

[Risks 5](#_Toc145261331)

[Controls 5](#_Toc145261332)

[Implementation Risks 5](#_Toc145261333)

[Infrastructure Providers 7](#_Toc145261334)

[Control 7](#_Toc145261335)

[Implementation Risks 7](#_Toc145261336)

[Network Configuration 8](#_Toc145261337)

[Controls 8](#_Toc145261338)

[Implementation Risks 8](#_Toc145261339)

[Environments 9](#_Toc145261340)

[Risks 9](#_Toc145261341)

[Controls 9](#_Toc145261342)

[System Design 11](#_Toc145261343)

[Risks 11](#_Toc145261344)

[Controls 11](#_Toc145261345)

[Implementation Risks 11](#_Toc145261346)

[Devices 12](#_Toc145261347)

[Risks 12](#_Toc145261348)

[Controls 12](#_Toc145261349)

[Data 13](#_Toc145261350)

[Data At Rest 13](#_Toc145261351)

[Data In Transit 14](#_Toc145261352)

[Risks 14](#_Toc145261353)

[Controls 14](#_Toc145261354)

[Infrastructure 15](#_Toc145261355)

[Data 16](#_Toc145261356)

[Data At Rest 16](#_Toc145261357)

[Data In Transit 17](#_Toc145261358)

[Authentication 19](#_Toc145261359)

[Authorisation 20](#_Toc145261360)

[Auditing 20](#_Toc145261361)

[Availability 20](#_Toc145261362)

[Auditing 21](#_Toc145261363)

[Appendices 24](#_Toc145261364)

[Appendix A - Document Information 24](#_Toc145261365)

[Images 24](#_Toc145261366)

[Tables 24](#_Toc145261367)

[References 24](#_Toc145261368)

[Review Distribution 24](#_Toc145261369)

[Audience 24](#_Toc145261370)

[Structure 24](#_Toc145261371)

[Diagrams 24](#_Toc145261372)

[Terms 24](#_Toc145261373)

## Introduction

Key qualities users expect to experience[[1]](#footnote-2) when using a system is being able to trust that they are free from risk. In essence, their assessment is based on an understanding that their use of the system will not lead to a divulging of confidential information.

This expectation aligns with In addition to the above desired quality outcomes, systems also have legal, regulatory and policy obligations. This includes respecting and protecting the privacy of citizens, whether external or internal to the organisation. This includes implementing specific technical controls to protect government data, depending on the information’s data classification.

## Outcomes

Ensuring Security of services is based on achieving outcomes -- not simply following processes.

The Outcomes

* Remain Available to end users wherever, whenever they need to use the service
* Provide Integrity of the data, such that it is not manipulable without authorisation, consent, and auditing.
* The Privacy of Users Information is respected and protected.
* Provide Confidentiality to Users while using the system

## 

The areas of consider whether sufficient controls have been impelemnted to mitigate the risks to an acceptable level can be described as follows:

* People: the stakeholders who are developing and delivering the service, and the users using the system.
* Delivery Artefacts: which may contain confidential contractual IP,
* Infrastructure: the physical protection of access to the server devices and other infrastructure
* Service Components: the components used to develop the service on the target infrasturcture
* Code Artefacts: the persisting of the works output, that may contain confidential IP and/or disclose areas of weakness in design of the various systems,
* Configuration Artefacts: the persisting of sensitive configuration information,
* Device & Service agent: the user’s means of accessing the service
* Data: the protection of the data

# People

Statistically, Delivery Stakeholders remain a key risk to the securing of confidential information.

## Risks

Examples of common sources of issues include but are not limited to:

* Responding to email based fishing attacks.
* Inserting hardware (e.g, thumb drives) that may contain malware.
* Inclusion of confidential environment access and/or system integration information (e.g. Service account Names & related Passwords, in documents, that when printed are outside the reach of automated controls.
* Committing resources containing confidential information to permanent stores documents (e.g., Confluence, Git repositories, GitHub, etc.) which have controls on them to disable removing them and/or give access to historical versions of the document prior to the removal of the confidential information.
* Stakeholders emailing confidential information to other stakeholders (e.g. offshore or simply external partners and consultants) across open public networks, via email text containing cut/pasted information, or unencrypted attachments (e.g. excel files containing collated confidential information).
* publishing to code repositories source code that contains confidential integration information (e.g. connection strings, service account credentials, etc.)
* being unnecessarily powerful Permissions within Prod Data environments.

## Controls

The primary means to mitigate delivery stakeholders being the source of leaked confidential information is by removing their access to confidential integration information.

This is achieved by

* mandatory security training of delivery personnel to understand
  + the types of information they have access to, therefore
  + they type of information they could disclose
  + how automation of deployment of systems and data can remove that risk
* developing automation of the deployment of system, configuration, settings and data.

## Implementation Risks

Risks to putting in place these controls include:

* not having in place a training process
* quality assurance stakeholders wishing to continue accessing systems directly, to do *manual* viewing confidential settings, rather than develop automation to test these settings.
* Quality assurance and/or other stakeholders arguing for and being given Roles and Permissions in production data environments to which there is no business service justification.

# Infrastructure Providers

Confidential information, whether system configuration information or user information, is maintained within Systems deployed to infrastructure maintained by providers.

Traditionally, providers were in-house.

Current best practice recommends first understanding cloud infrastructure possibilities and constraints, then using accreditable cloud providers such as Microsoft Azure or Amazon cloud services.

## Risks

The use of infrastructure provision services that do not have sufficiently rigorous controls in place to ensure their availability and security.

## Control

The minimum control is for the infrastructure provider to have received ISO-27001 level 2 or better from a third party.

ISO-27001 Level 2 ensures a range of controls, starting with Personnel evaluation and training, through to ensuring physical and virtual access to premises, infrastructure, components and data is restricted, authorised, and audited[[2]](#footnote-3). Redundancy of service is by design a default, from power supply upwards.

## Implementation Risks

Risks include:

* the infrastructure service provider reporting ISO-27001 compliance based on a self-assessment process (ie, achieving only Level 1) rather than using an independent 3rd party service (Level 2).
* The provider not disclosing the findings of remaining unmitigated risks found during the development of the ISO-27001 Stage 2 report.
* Not understanding and/or utilising cloud options correctly and deploying Services as VMs to Infrastructure as a Service, increasing operational costs, reducing budget for maintenance and improvement. Reduction of maintenance budget reduces oversight and mitigation of technology risks that emerge during the service’s lifespan.

# Network Configuration

Services deployed to device infrastructure within a network has the potential of being compromised. A compromised device may have access to unsecure devices with which it communicates, and/or unsecured devices in the same network.

## Controls

The primary control is to develop networking subnets for key system services & devices, such that they have access to only the devices they need to communicate with, and only let in traffic from appropriate devices in the same system.

For example,

* Storage devices or services (eg the system’s relational data base service, such as Sql Server or Postgres) must be within a subnet such that it can only be accessed from
  + the service’s logic device, and
  + the data backup and restoration automation
* The logic tier device (e.g. the web server) should be protected within a subnet such that it can only be accessible from the web, and by system configuration and settings automation. The subnet should not permit communication to any other device or service than the ones it utilises (caching, data storage, IdP, etc.)
* Where there is choice, configure subnets to only accept secure communication between system devices (e.g., 443 instead of 80, etc.) and deny insecure communication between system devices.

Note that for a better user experience, subnets around web servers accessed by human end users (as opposed to APIs web servers accessed only by machines) it is preferable the service accepts insecure connections (port 80) but redirects them to their equivalent secure (port 443) endpoint.

## Implementation Risks

Risks to this being implemented correctly include:

* No deployment or configuration automation being used. This could indicate a poor skillset that will in turn argue against the complexity of subnetwork based controls.
* Providing physical access to devices to stakeholders (e.g. quality assurance specialists), in doing so, compromising the confidentiality of integration credentials.

# Environments

Projects develop multiple environments prior to making services available to end users. The list of environments commonly includes most of the following:

* Built Test (BT), for deployment pipelines to run automated static and dynamic work item acceptance tests,
* System Test (ST), for Quality Assurance specialists to do exploratory investigations to inform their decision as to what further automated tests to develop,
* User Test (UT) for non-delivery stakeholders to access
* Compliance Testing (CT) for 3rd parties to test their API service clients with.
* PreProd (PP) for testing prod integration configuration,
* Prod (PR/OD) for End User accessible deployment for collection and management of production data.

Other environments shared between one or more environments include:

* BACKUP for backing up PR and PP.
* TESTBACKUP for backing up data that is non-prod

## Risks

The key risk is that processes are immature and Production data is used in any other Environment than PR/OD, and backed up to any other environment than [Prod data] BACKUP.

It is common for immature organisations to believe that the use of older prod data (eg, a month old) is acceptable in ST. This is incorrect. Unauthorised access to an environment containing data that is a month old is exactly the same as being hacked in a production environment, a month ago).

## Controls

The risk can be removed by

* personnel training (see earlier) as to the tremendous danger it
* developing Testable Data suitable for use by testers and other stakeholders in a non-prod environments (all environments besides PROD).
* Developing automation that deploys and configures non-prod environment devices to connect to non-prod databases.
* This includes PP, where it deploys, configures and tests the connection to a Non-Prod Database deployed to the Prod Data environment.

# System Design

## Risks

The key risk in regards system design of cloud hosted services is the use of an inappropriate system design pattern. Specifically, in the case of cloud architecture, there is a risk to using the *Microservices pattern*, and in doing so significantly increasing delivery risk for a benefit that is not applicable to most organisations.

Specifically, the Microservice pattern[[3]](#footnote-4) significantly increases the number of devices and services that require securing, monitoring, and maintaining, while *also* increasing the breadth and complexity of the delivery automation required to these devices and services.

## Controls

Unless a warranted outcome of decision by project governance, avoid a system design that differs from a classical *modulithic*[[4]](#footnote-5) approach, such that the system design is easier to understand, implement, deploy, monitor and iteratively improve.

## Implementation Risks

Developers are susceptible to soft and hard marketing as much as anyone else. Marketing by an organisation that markets infrastructure, such as Amazon or Microsoft, to use more of their products is an constant issues that adds risk to design decisions by either developers or project governance members.

# Devices

Cloud infrastructure is notable for providing infrastructure platforms to suite current demand, hence providing IaaS, CaaS, PaaS.

While there is plenty of advice and evidence to avoid IaaS and instead use a PaaS approach, demand for IaaS continues.

## Risks

The use of IaaS is approximately 3x[[5]](#footnote-6) the cost of on-prem managed infrastructure. The increase in infrastructure costs removes budget from BAU operational and system maintenance and improvement opportunities. The increased cost also decreases the benefit/cost equation, therefore reducing the lifespan of the service.

## Controls

On the other hand PaaS based services can be delivered fro approximately 1/3 the cost of on-prem managed infrastructure. The preferred control to decreasing infrastructure, development and maintenance costs is to ensure the development and deployment specialists are fully versed in cloud architecture options and experience in PaaS, to understand the constraints and workarounds necessary to take advantage of its its cost savings.

# Data

The data associated to ICT systems is categorizable as either being at rest or in transit.

### Data At Rest

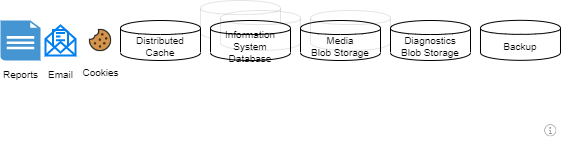


Figure : Data Storage Means to Protect

This category includes:

* On servers:
  + System Databases, which in turn can be
    - relational (e.g. Sql Server)
    - or non-relational (Blobs, Tables, Documents, Graphs)
    - discovery service (e.g. elastic search services) data,
  + Shared Cache,
  + Files, including temporary Diagnostic logs,
  + Code bases in managed Code Repositories
  + Backups of one or more of the above.
* On Client Device:
  + Browser content cache,
  + Cookie storage,
  + Certificate storage
  + Clones of repositories
  + Emails received from the system
  + Downloaded system generated Files (eg reports)
* In Client environment, outside of the devices:
  + Printed Emails
  + Printed Reports
  + USB Keys containing copies of downloaded files.

### Data In Transit

This group includes:

* + Request and response between client and service devices
  + Integration messages between service devices (server, cache, data storage, Identity Providers, etc.)

## Risks

A number of risks can compromise the Confidentiality Integratity and Availability (CIA) of the data.

## Controls

Confidentiality of Data in Transit is addressed by:

* Requiring channels between all devices to be encrypted when technically achievable and reasonable. This includes between the end user’s browser and the service (using HTTPS), and between service devices.
* Redirecting connections across unencrypted channels (eg port 80) from end user browsers to encypted equivalent endpoints (eg port 443).
* Refusing connections across unencrypted channels between service devices.
* Relying on non-password-based service accounts (eg: MSI on Azure) between devices
* Where not available, using credentials kept in an secure, audited, project specific secure key stores only accessible by service accounts used by automation pipelines (so that no human stakeholder can view the data).
  + It is important to use a project specific keystore to limit the number of stakeholders who have access to view the credentials

Confidentiality of Data at Rest is addressed by:

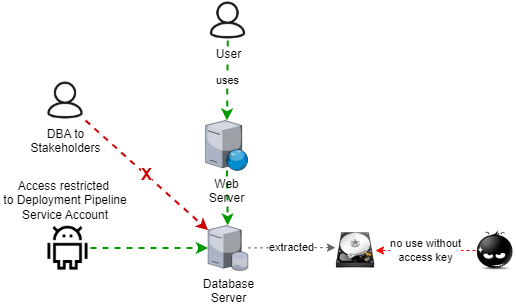


Figure : the value of Encrypted Data is very low.

* choosing technologies that permit encryption at rest.
* Turning on encryption of storage when it is technically available and not impractical.
* Using a keystore managed key for the encryption,
* Hosting data on infrastructure that is physically unable to be accessed, all interactions with it are audited (e.g.: Cloud based services), to which downloading of backups or similar is limited.
* Do not using file based diagnostics logging.   
  File based diagnostics logging is no longer practical on dynamically allocated cloud based infrastructure, and best practice is to use a cloud based target (eg. Blob storage).

#### Infrastructure

The infrastructure is developed by that can be independently validated to:

* ensure encryption is enabled when it is a reasonable option
* refuse unencrypted access when it is a reasonable option
* relies on non-password-based service accounts (e.g.: MSI) before falling back to traditional service accounts, enabled by using credentials retrieved from secure key stores, falling back to using credentials directly, again retrieved from secure key stores
* that do not publish (e.g. in log files) the credentials

#### Data

#### Data At Rest

Data at rest comes in many forms:

Data & Backups

System data and backups are protected at rest by:

Blob Storage

Media (photos, pdfs, etc.) is not stored in relational databases but instead in Blob Storage.

Enabled encryption of these datastores is a standard feature of cloud storage providers.   
  
A review of the infrastructure development code during a white box security test would be sufficient to demonstrate this has not been disabled.

Diagnostics Storage

Storage of diagnostics log files to file systems is no longer practical in cloud infrastructure environments, especially on dynamically allocated infrastructure. The preferred current approach is to send messages to blob storage – encrypted as per media blob storage described above.

Cache

Although volatile, memory caches can be encrypted in some cases, depending on the technology offered by cloud vendors. It is an option, depending on the confidentiality of the information cached.   
  
The nature of the data persisted by this service is not deemed sufficient to obligate this outcome.

Tokens

Important data – Authentication and Authorisation Tokens – is persisted on Service Clients.  
To not permit bleeding of internal system architecture information or user data, the tokens must be encrypted, using *Secure* flags.

Messages

Generated Messages may disclose information to which access cannot be audited when they accessed outside the system (e.g., as email). Examples of Confidential information that should not be disclosed is Personal Identifiers.   
  
Note: Training (see above) is required to assist Business Support and Operations Stakeholders to develop Message Templates with this concern in mind.

Reports

Generated Reports that can be printed and accessed outside the system can disclose lots of information to which access cannot be audited. Examples include lists of Personal Identifiers (NSNs, etc.), Scores, etc.

Principles direct system design choices to prefer ensuring reports can only be accessed in-system, and/or exclude Personally Identifiable Confidential Information.

Note: Training (see above) is required to assist Business Support and Operations Stakeholders to develop Report Templates.

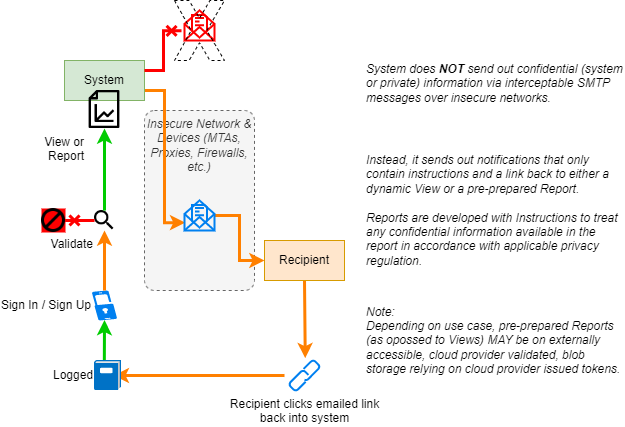


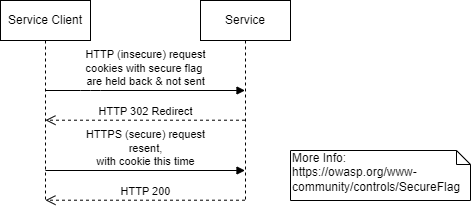
Figure : Audited Access to Report based Static Data

#### Data In Transit

Data connections between logic devices (e.g., webservers) and storage devices (e.g., database servers) are encrypted.

HTTP Redirection

Unencrypted connection attempts from service clients are redirected to encrypted connections via HTTPS Redirection.



Note: API Service Clients must include logic to correctly respond to 302 redirections.

Note:  
HTTP Redirection of custom and SaaS products is tested during independent security testing.

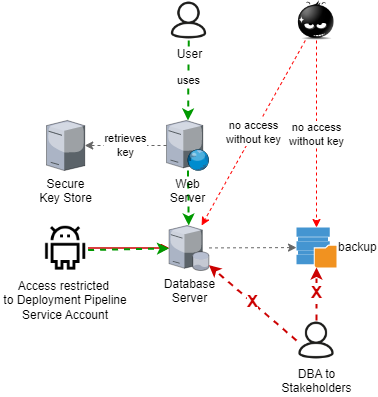
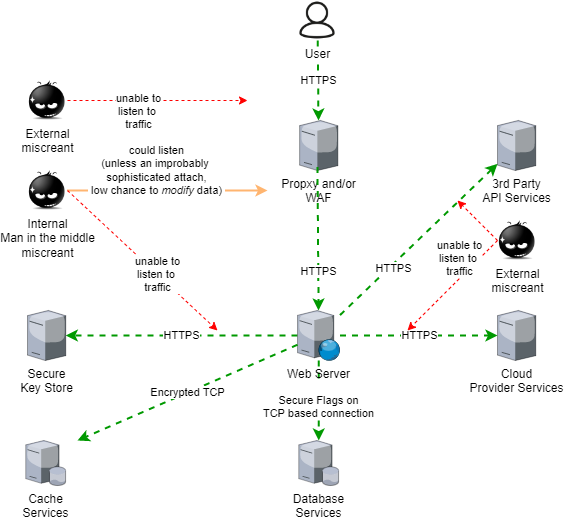


Figure : Encrypted datastore connection strings

Service Vendor Responsibilities: note that providing security of data at rest and in transit is a deliverable quality of the contracted SaaS service, but how they do so is beyond the expectation, delivery, and verification scope of this project.

Encrypted Integration Channels

Encryption of communication between all service components are maintained to use current recommend channels (TLS 1.3+).



Note that verification of vendor SaaS product component integration is out of the delivery scope of this project unless contractual.

#### Authentication

All System Users are associated to Sessions as either Non-Authenticated or Authenticated Users.

Non-Authenticed users start a session, enabling meaningful logging, but are redirected to the IdP to enable being converted into Authenticated Users.

It is a Principle, and Requirement, that Systems use a dedicated secure 3rd party Identity Provider (IdP) rather than store credentials in-system, as described in the Integration View.

Note: some systems – e.g., the Integration Information Portal – permit both Authenticated and Public non-Authenticated access to information.

#### Authorisation

At a minimum, all Operations that can manipulate production data must be authorised against Policies – generally Policies that require the current Session’s system User to have specific Permissions (as bundled in their assigned Roles).

### Auditing

In all service systems, all operations that can manipulate production data are audited, recording the current Session’s Authorised and Authenticated User.

Succinct Auditing information is required to be kept for a timespan (e.g.: a configurable 90 days) -- preferably permanently.

#### Availability

##### SLAs

Service components are hosted on Cloud Infrastructure that provide a 99.99% per month availability.

Vendor services are procured based on Requirements to meet acceptable SLAs.

Disaster Recovery processes are developed, as pr the Maintenance View.

##### Horizontal Scaling

Dynamic Horizontal Scaling Availability is a quality of the contracted SaaS service, and therefore by definition beyond the delivery scope of this project, but it can be reported the service’s infrastructure is dynamically horizontally scalable to meet peak needs:

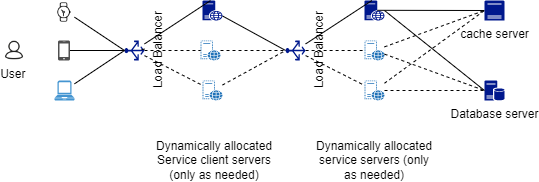


Figure : Dynamical Horizontal Scalability

##### Session Concurrency

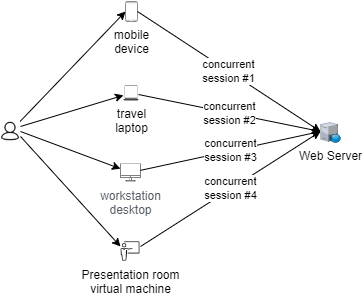


Figure : Sessions can be concurrent

To accommodate current usage patterns Sessions can be Concurrent.

#### Auditing

Operation Auditing is provided as a feature of the contracted SaaS Services, therefore outside delivery scope.

But it can be reported that its Operations are audited, associated to Sessions assigned to User on a unique Device.

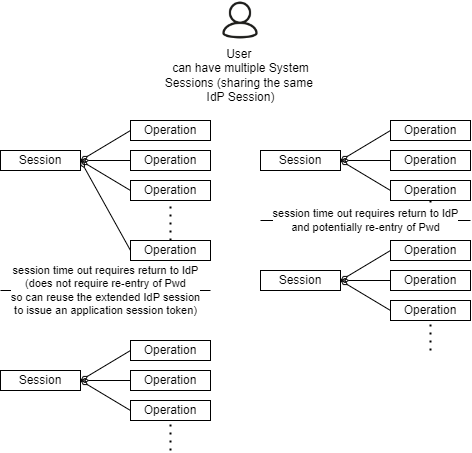


Figure : Operations are Audited per Session (per User per Device)

Appendices

Appendix A - Document Information

### Images

[Figure 1: TODO Image 2](#_Toc144995112)

### Tables

[Table 1: TODO Table 3](#_Toc145048484)

[Table 2: TODO Table 2 3](#_Toc145048485)

### References

**There are no sources in the current document.**

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
|  |  |
|  |  |
|  |  |

### Audience

The document is technical in nature, but parts are expected to be read and/or validated by a non-technical audience.

### Structure

Where possible, the document structure is guided by either ISO-\* standards or best practice.

### Diagrams

Diagrams are developed for a wide audience. Unless specifically for a technical audience, where the use of industry standard diagram types (Archimate, UML, C4), is appropriate, diagrams are developed as simple “box & line” monochrome diagrams.

### Terms

Refer to the project specific Glossary.

##### Term

: the meaning.

1. See ISO-25022: Effectiveness, Efficiency, Satisfaction (Inc. *Trust*), Freedom from Risk, Context Coverage [↑](#footnote-ref-2)
2. For example: no personnel may enter the data environment alone for any reason (e.g.: to do physical work) without being supervised by a peer who records their operations. [↑](#footnote-ref-3)
3. MicroServices is a pattern to work around different departments being unable to coordinate their work items towards delivering a single service. While very large companies such as Amazon do have a need for large and separate delivery teams working on the same service (eg: for Discoverability, Item Presentation, Shopping Experience), the vast majority of organisations do not, having only one department. [↑](#footnote-ref-4)
4. Modular feature sets using a common monolithic base. [↑](#footnote-ref-5)
5. The cost increase would be even more if on-prem processes were required to be improved to an equivalent ISO-27001 level of compliance as cloud providers offer. [↑](#footnote-ref-6)