IT Project Guidance

Design:  
Integrations – Common Defaults

Version:

0.1

## Description

While every service is presumably for a novel business purpose, and the order in a project’s work items are undertaken may vary slightly from project to project, the underlying supporting services are more or less the same, project to project.   
This document outlines and describes the purpose of these default services, to be considered when developing system designs.

## Synopsis

All systems require assistance from dependency services to manage identities, store resources (records, documents, media, metadata) in datastores, find them again with dedicated search services, serve them quickly using caching, and be able to notify subscribers to events and changes that may be important to them.

## Contents

[Description 1](#_Toc151362676)

[Synopsis 1](#_Toc151362677)

[Contents 2](#_Toc151362678)

[Introduction 4](#_Toc151362679)

[Risks 4](#_Toc151362680)

[Resolution 4](#_Toc151362681)

[Integrations 5](#_Toc151362682)

[Inwards 5](#_Toc151362683)

[Enterprise Website 5](#_Toc151362684)

[Dedicated Brochureware website 5](#_Toc151362685)

[A Self-Help Documentation site 5](#_Toc151362686)

[An Assisted Support Service 5](#_Toc151362687)

[Service Client 6](#_Toc151362688)

[Outgoing Dependencies 6](#_Toc151362689)

[Identity 6](#_Toc151362690)

[Data Storage 6](#_Toc151362691)

[Confidential key store 6](#_Toc151362692)

[Caching 7](#_Toc151362693)

[Relational Database 8](#_Toc151362694)

[Document Database 8](#_Toc151362695)

[Graph Databases 8](#_Toc151362696)

[Search 9](#_Toc151362697)

[Notification Services 9](#_Toc151362698)

[SMTP Email Service 9](#_Toc151362699)

[Pop / IMAP Email Service 10](#_Toc151362700)

[Public Twitter / X / Mastodon / etc. 10](#_Toc151362701)

[Service Monitoring Services 10](#_Toc151362702)

[Client Behaviour Monitoring Service 10](#_Toc151362703)

[Infrastructure Monitoring Service 11](#_Toc151362704)

[Functionality Control Services 11](#_Toc151362705)

[Feature Control Service 11](#_Toc151362706)

[User Services 11](#_Toc151362707)

[Feature or Resource User Rating Services 11](#_Toc151362708)

[Information Services 12](#_Toc151362709)

[IP-Geo Conversion Service 12](#_Toc151362710)

[Mapping Services 12](#_Toc151362711)

[Decision and Workflow Management Services 13](#_Toc151362712)

[Rule Engine Services 13](#_Toc151362713)

[Workflow Management Services 13](#_Toc151362714)

[Appendices 14](#_Toc151362715)

[Appendix A - Document Information 14](#_Toc151362716)

[Versions 14](#_Toc151362717)

[Images 14](#_Toc151362718)

[Tables 14](#_Toc151362719)

[References 14](#_Toc151362720)

[Review Distribution 14](#_Toc151362721)

[Audience 14](#_Toc151362722)

[Structure 14](#_Toc151362723)

[Diagrams 14](#_Toc151362724)

[Terms 15](#_Toc151362725)

## Introduction

While each system is presumable built to meet a distinct business case[[1]](#footnote-2).

## Risks

While there are plenty of developers, there are far fewer who have developed and designed systems, especially for national services.

The risk is that developers will begin working on what they know, or work on front end work items to develop front end views to have something to present to business stakeholders and keep them engaged, when structurally, a systems maintainability, modularity and flexibility benefit from being built from solid foundations upwards.

## Resolution

This document outlines for non-developer stakeholders the dependency services that services will integrate with, so that they there is a better understanding of why they are in the design, and possible why there sometimes work is required on them before doing work that is more apparent to business stakeholders.

# Integrations

Integrations can loosely be categorised as inward only, outward only, or both directions.

## Inwards Integrations

While inward integrations are not dependencies, inward integrations are worth touching on lightly here so that they are not omitted from planning.

### Confidentiality

#### Certificate Services

Another service that is tightly related to delivering services while not being directly integrated to a service or its parts is the development of HTTP/S backing certificates that expire and need replacing regularly[[2]](#footnote-3) to provide end users security.

Aspects to consider include:

* being aware of the enterprises domain naming strategy,
* having the means to organise and register the development of domain name service record entries (see Domain Names Service(DNS), under Discovery).

### Integrity

#### Web Access Firewall (WAF)

WAFs are used to protect one or more systems behind a rule system that controls and limits traffic from/to known IPs and ports. WAFs also protect systems from malware[[3]](#footnote-4).

It’s an incoming relationship/link to one or more of the following services (Enterprise website, brochureware website, Self-Help site, assisted support site, API Gateway, service site, etc.).

Aspects to consider include:

* meeting with maintenance specialists of these services and communicating the downstream URLs to the allowed.

#### Firewalls

WAFs may or may not provide firewall functionality. If not integrated, a firewall service may require configuration to permit traffic from certain locations to the business service or its supporting services.

Tasks to consider include:

* meeting with the maintenance specialists of these services to inform them of your requirements, services components, etc. for them to develop permitted routes.

#### API Gateways

API gateways provide a way to monitor, audit, and provide better discoverability of the APIs of multiple business services[[4]](#footnote-5).

Aspects to consider include:

* Media (descriptions, images, etc.) to be published on the API gateways static pages required developing,
* The API documentation site requires being linked to this gateway.

### Discovery

#### Domain Naming Service

Users can’t find services on the net by IP alone – especially if the IP addresses are non-static. Instead, they rely on domain name service (DNS) records so that they can records that translate their typed address (*yourorg.tld*) to the current IP for the service or its related services (self-help documents, etc.).

#### Enterprise Public Website

The enterprise’s public presence website will presumably have a small number of pages describing the service and linking to it.

Having something external to the service describe the service, in a SEO manner, and then point to a service, improves its discoverability, decreasing the need – but not removing it completely – for comms and engagement to be worked on.

Aspects to consider include:

* Meeting with the specialists in charge of the enterprise website to understand their processes and constraints for developing media, submitting it to them, and coordinating their publishing date.

#### Enterprise Intranet Website

The enterprise’s internal website (e.g.: Confluence or similar) will also be a place where a subsite will be created, which in turn link to the service and its dependent sites.

Aspects to consider include:

* Making one or more deliver project member(s) responsible for developing and maintaining the text on the website, keeping links current, etc.

#### Dedicated Public Brochureware Website

In some cases, the service will benefit from a dedicated brochureware site that is separate from the enterprise website.

These sites help unify the user experience, linking together distinct services that are logically but not technically integrated. Examples include the service itself, a self-help documentation site, an assisted service site, an APIs documentation site.

Using an external website not only improves the end user experience by providing a comprehensive and unified experience, it also improves its portability so that with only a couple of esthetical changes (logo, title, subtitle, etc.) it can be sold or handed off to a 3rd party service provider to manage on the enterprise’s behalf.

Aspects to consider include:

* Understanding the domain naming scheme used.
* Meeting the specialists in charge of putting up custom micro websites, and understanding their processes and constraints for developing media for publishing.
* Assigning to one or more project delivery stakeholders the development of media to be published on this website, and maintained current.

### Learnability / User Assistance

#### A Self-Help Documentation website

It’s a bit uncommon due to technical issues[[5]](#footnote-6), but the self-help documentation *may* link to specific pages on the site, so qualifies as an incoming integration.

Aspects to consider include:

* Selecting a service that doesn’t require users to authenticate themselves,
* but capable of offering Singe Sign On (SSO), shared with the service itself,
* that can accommodate and assist users to find culture and location specific documentation (e.g.: en-NZ, mi-NZ).
* Can be configured to be accessed via a project specific URL (e.g.: *help.ourservice.ourorganisation.tld*).
* Provides deep linking to specific pages and sub headers on them. Optionally highlights specific search phrases.
* May collect feedback on the self-help documents themselves to improve their quality as well (not just the service).
* Registering the subscription’s expiration and re-subscription needs on both the project risk and recurring event registries.

#### An Assisted Support Service

An assisted service may be linked to from a dedicated brochureware website, and may link to the service website, but again this will be in a limited basis for same reason as the technical difficulties that self-help documentation sites run into.

Aspects to consider include:

* Selecting a service that can provide multi-cultural user experiences (e.g.: en-NZ and mi-NZ)
* Selecting a service that can be configured to be accessed via a project specific URL (e.g.: *support.ourservice.ourorganisation.tld*).
* Selecting a service that provides deep linking between emails and latest ticket threads, etc.
* Arranging the funding and subscribing to a SaaS service account
* Selecting a service that doesn’t require sign in to start a session, but that authentication is required before ticket resolution so that auditing is occurring appropriately,
* Registering the subscription’s expiration and re-subscription needs on both the project risk and recurring event registries.

### Reportability

Users put information into systems in order to then retrieve, summarise, and report on to in turn support decisions and actions.

#### Reporting Database

Because reporting is surprisingly resource intensive in terms of CPU and IO so it is common practice to develop a second database specifically for reporting needs.

The exporting of the data from the system’s transactional database to the Reporting database is done via an ETL engine.

Note:  
ETL operations used to be done almost exclusively database to database. This approach is now considered poor practice as it skips validation on the target database and introduces inconsistencies that are hard to track down and correct.   
Where possible, use ETL to invoke APIs of each system.

That said, since reporting databases generally don’t have an distinct service with APIs wrapping them, there is usually no other option than using ETL from/to database.

The reporting database will still be accessed from interfaces within the service (but may also be accessed from an external dedicated reporting service (e.g.: Microsoft BI) discussed next.

#### Data Analysis Service

While basic preformatted reporting may be offered to authenticated users, it may be required that others have the ability to develop ad hoc queries and developing complex custom reports.

Aspects to consider include:

* Ensuring data that the analysis service has access to does not contain Personal Identification Information (PII) or even personal information (PI).
* Ensuring the reporting is using a dedicated reporting database so that requests, especially custom developed ones, do not impact system users experience.

### Usability

#### Service Client

Although self-explanatory, a service agent (i.e., a browser) will be used to establish a connection with the service and render an interface for end users.

Aspects to consider:

* Communication between the devices must be secured against eavesdropping. On the HTTP based section of the WAN, this is done via using HTTPS instead of HTTP.
* To not leave an insecure route open, permanently redirecting all unsecured HTTP traffic towards an HTTPS equivalent.
* The establishing of a secure comms channel by installing a cert, and the redirection rules can be requested of Maintenance specialists to add to the Infrastructure as Code instructions pushed to develop and maintain the underlying web server, or on a WAF in front of it, or wider solution (e.g., Cloudflare).
* Because insecure Cookies containing confidential information – e.g., a session cookie containing a session identifier -- may be sent over the unprotected HTTP channel before they are redirected back to a secure HTTPS equivalent, they have to be marked ‘secure’.

Note:  
Marking them ‘http-only’ at the same time is also recommended, but for different reasons (it limits JavaScript running within a browser from accessing the contents of the cookie).

## Outward Service Dependencies



Figure 1: Outgoing Integrations to Dependency Services

### Identity

Identity Provider (IdP) services are an essential

### Data Storage

Data storage used to come in one size: relational databases. Options have greatly expanded over the past 15 years, providing specialised solutions that each have their role to play.

#### Confidential key store

The use of confidential key datastores are current best practice for persisting confidential system integration credentials.

Confidential key stores have features such that once entered, control panel GUIs don’t show the key value and block copy/pasting it back to desktops. Accesses are audited and logged for later review.

Note:  
In legacy designs, a system config file (e.g.: config.xml) might have been the recommended place to persist integration information.

The approach of using a config file had several security impacting limitations: as the information was in a code file, it often got checked into a code repository, with the inability of clearing it out once discovered, essentially compromising system security. If encrypted using a device MAC or similar, it required all servers to be configured with the same MAC, which is impossible to implement on shared servers, PaaS, etc.

Credentials that are stored in a keystore include the services described in this paper - namely:

* Data storage:
  + Relational database connection string identities
  + Document database credentials
  + Graph databases
* Caching:
  + Caching service credentials
* Search:
  + Search API credentials
  + Ip-location geolocation API credentials
* Etc.

#### Cache

Caching Services are in-memory key/value datastores shared between servers. They provide a necessary work around to the bottleneck of other forms of datastores (relational databases, etc., described next) that interact with underlying physical hard drives and are orders of magnitude slower than in-memory operations.

Note:  
In this list caching is purposely put before other forms of data storage simply because if not implemented early, developers don’t get into an habit of requesting resources from cache first, only falling back to datastore queries second, when the cache item has timed out, or it’s simply not appropriate to cache. Caching is one of those cross-cutting services that takes a tremendous unnecessary amount of effort and cost to retrofit in correctly later.

#### Relational Database

For decades the relational database has reigned supreme as the most practical and efficient means of persisting data[[6]](#footnote-7).

While ‘old’ compared to more modern forms of data storage (NoSQL, etc.), it still rightly holds its primary position[[7]](#footnote-8) in critical systems.

Partly due to its age, it doesn’t use HTTP/S to accept request, it uses other ports  
(SQL Server default port is 1433).[[8]](#footnote-9)

The credentials required for developing a connection string to query this service are either retrieved at startup from the credential store mentioned earlier, or better yet, one uses password-less service accounts (e.g., Microsoft Secure Identities).

#### Document Database

While current relational databases are now capable of also persisting document records, there are often reasons to persist data in databases services that are specialised in document storage.

Use cases for document storage include the storage of documents, tagged with non-relational database.

Note:  
document databases have several advantageous and disadvantages compared to relational databases. Advantages include ease of use, as well as the ability to get going when a relational schema has not been defined, and/or standards are still evolving, requiring a flexible storage format. Disadvantages include the requirement to perform schema validation in the logic layer (as it is not performed at the data layer), as well as the high cost of multi-record updates when relational patterns become apparent later.

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

#### Blob Storage

Relational databases can be used but are a poor choice for where to persist uploaded media[[9]](#footnote-10).

blob storage is best suited for storing streams

#### Graph Databases

While fringe 15 years ago, graph databases are front and centre data stores in current designs. They are the basis of establishing links between resources to develop context (parent, children) and sequences (previous, next, and next choices), as well as relationships (equivalencies, opposites, recommendations, etc.).

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

#### Search

Similar in some regards to cache, Search services[[10]](#footnote-11) are another type of high-speed mostly-in-mem database (they are also backed by storage).

Most forms of relational or non-relational database services suck at providing a means to search through them in the way that users use them in the real world. For one users mistype terms all the time, spelling phonetically, for another they also forget the exact term and write synonyms for a term they are looking for, and finally, they expect it to be blindingly fast, usable for auto completes, or returning values while they are still typing.

This essentially is everything that traditional databases, which focus on being good at other things, are not especially good at.

Hence the need for a specialised in-memory service that are being updated when records are being stored in traditional databases, making search tokens phonetically, etc.

Elastic search is a market leader in this space offered by all the major cloud service providers.

Basic characteristics of valuable search service include:

* Takes written search term
* Forgiving of spelling
* Forgiving of spelling in different cultures (for example Māori, “wh” is closer to an ‘f’, whereas in NZ English it’s closer to maybe “wer” or “wo”).
* Capable of finding information via synonyms if configured to do so.
* Searching by qualities (e.g.: metadata) describing resources (e.g.: filtering for images as opposed to sounds or text)
* Rapid
* Inexpensive as it will be used a lot.

Additional characteristics to seek within search services include:

* Developing personalised search, using previous search patterns of the identified user to influence the scoring and ordering of returned data.
* Reporting is available to report on:
  + which search terms were used the most,
  + which search terms did not produce usable results, as can be seen by immediately following it with a similar but different search clause.

At this point in time (Q4/2020) consider the value of the following cloud services to your service:

* 3rd party services: Elastic Search, Algolia, Other
* Custom: Azure Search, Amazon

Thumbs up sign  
**Tip:**  
In order to get to MVP quicker, it is highly required that the project does not develop a system specific search engine (e.g.: Azure Search) unless service costs due to high usage warrant looking into it.

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

### Notification Services

Users, non-users, system components and collaborative systems require being kept up to date on state and system changes.

#### SMTP Email Service

Every system needs to send out notifications to subscribed consumers.

In may be that some dependency services – e.g., IdPs – will handle their own email notification needs, but every mature system evolves to needing to send out its own notifications.

This may be to inform users that a system may be unavailable for a duration of time starting on a specific date and time, due to an upgrade.

This may be to inform users that a new major release is available, with a list of new features.

This may be to inform users that a disclosure statements (e.g., tracking, privacy, data use, terms & conditions, etc.) of the service have been updated and the latest version can review and accept by them by clicking a specific link, etc.

The service used may be either an internal enterprise service (e.g., Microsoft Exchange, Google Apps, etc.) or a dedicated one-way service (Mailchimp, etc.).

Note:  
It is highly recommended that the domain used by the mail service as the sender domain comply with DMARC protocol and similar requirements – anything to improve the credibility and recognisability of the sender.

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

#### Pop / IMAP Email Service

There are very few cases where a service needs to use IMAP or POP to query an email service to get a listing of email messages waiting for a person. But it can happen.

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

#### Public Twitter / X / Mastodon / etc.

There may be some cases where a system needs to integrate with a social media service to publish notifications to the world and not just its users (as email-based notifications are limited to).

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

#### Service Bus Service

Whereas SMTP and X are services intended to notify users, a pub/sub service is used to publish and subscribe messages to decouple components of a system. It is an essential aspect of highly maintainable systems that are used over a long service lifespan.

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

### Validation Services

#### Malware Detection Service

It’s practically an enterprise obsession that all services are protected by a shared WAF service.

While this helps contribute to a defence in depth posture, it doesn’t remove the need for a service to defend itself.

The most direct route to introducing malware to a server is by allowing users to upload media. It is not appropriate to disable or not provide this functionality because it is a foundational capability, permitting users to customise their user identity’s avatar, upload personal backgrounds, pdf and word documents for collaboration and/or publishing, etc.

Secondly, whereas a WAF can spot known malware, there is always a risk that the WAF malware signature database was not 100% current and a malware got uploaded and persisted, and from them on will infect downloaders of the media. For this reason, it is very important that a malware service be integrated directly into a system so that it periodically rechecks stored media for malware using a periodically updated virus definitions dictionary. It pays to recheck as it is far cheaper to subscribe and integrate such a service than the reputational damage from being the source of malware.

### Service Monitoring Services

Maintenance Specialists can’t keep a system running if they have no insights or notification as to unexpected behaviour and resource consumption before users call it in.

#### Client Behaviour Monitoring Service

For a decade or two, a lost effort quick win was the addition of Google Analytics to client-side code to capture user consumption habits (what pages were visited, by whom, from where, which links they used). Current privacy regulation around the use of cookies for data tracking has made this practice optional, therefore far less valuable.

Current best practice is to avoid the use of cookies and use serve side integrated services to track usage, by whom, from where, and what links they touch.

Note:  
The data collected does not include server-side behaviour, but that capabilities can either be optional, based on the user accepting cookies for that purpose, or still be exercised if avoiding the use of cookies by calling back home to dedicated APIs[[11]](#footnote-12).

#### Infrastructure Monitoring Service

Most infrastructure services – including cloud providers of course – provide services to monitor the infrastructure and services that has been subscribed to.

When correctly configured, the information is helpfully collected for viewing user interfaces for charting, monitoring, notifying, etc.

Information that is of importance include service qualities (requests/time, throughput, time to complete), and resource qualities (memory consumption, CPUs consumption, I/O to datastores, number of devices) to better inform vertical and horizontal scaling, preferably dynamically.

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

### Functionality

#### Feature Control Service

Long running services will require larger changes that will impact users, that potentially could be performed in one of several ways, with no clear path as to which one is better.

A common way to test the options with users is to integrate and depend on a feature flag control service that allows a subset of users, e.g., those located in a specific area (hence the need for an IP to geo services mentioned later).

Most cloud services provide such a service.

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

### User Services

#### Feature or Resource User Rating Services

For users to rate system features and resources hosted on them, a rating system has to be integrated, that provides a button to end users, that provides a link to an external integrated service, passing it an identifier for the page and/or resource.

Note:  
Being relatively trivial, one can be tempted to build this functionality directly into a service, but it is still recommended to subscribe to a service that provides the functionality needed to rate a service (either by stars or smileys) and potentially comment on it, then providing the means to moderate comments before and/or after they are posted. Additionally, these services can provide integrations with other services (assisted support, public recommendations, public visibility of work items, etc.)

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

### Information Services

#### IP-Geo Conversion Service

In enterprises there is a fascination with letting WAFs Geo-block requests from certain countries[[12]](#footnote-13). For this reason, geo-blocking is not seen as an application capability, and hence adding a IP-geo conversion service is often not added or added much later.   
We recommend that knowing where your service consumers come from, at what times, is an essential analysis concern, and instead recommend adding it early.

There are several services one can consider. Not many are free or offered by cloud providers, so a 3rd party subscription is required – which implies the expiration of accounts and/or payment methods needs to be put on both the risk and recurring events registry of the delivery program.  
  
The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

#### Mapping Services

Another service that mature systems seem to evolve to relying on is a mapping/postal address services, which are useful for improving data quality by autocompleting billing and shipping address entries, finding the postal code for an address, find the geolocation of an address, distances between points, placing GPS values on a map, etc.

Note:  
For other metadata about addresses, e.g., voting blocs / voting wards, school districts, etc. other services are required, but at least a mapping service can ensure the address is correct and accurate before launching a request for more information based on them.

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

### Decision and Workflow Management Services

#### Scheduling Services

A system can be scheduled to do work by an external dedicated scheduling service if it has the APIs to do so.

Common scheduling may include scanning users for collecting information on whom to send notifications to (welcoming new users, pinging users that have not signed in lately, not used a specific feature, etc.).

Scheduling can also be used to synchronise migration of data to the reporting database or to a data warehouse.

Aspects to consider:

* The cloud-based scheduling service may require an on-prem agent that can be invoked remotely from a cloud-based scheduling service.

#### Queue Services

Operations that are not required to be processed in real time can be offloaded. There are plenty of services that do this work well.

#### Rule Engine Services

Using external rule engine services provide a means of offloading rule processing to specialised services. Using a Domain Specific Language (DSL) to define rules has some value: they provide a means of encapsulating logic separately from the rest of the service and its dependencies by requiring dependencies to logic being worked out first, as they only deal with inputs and outputs (e.g.: it doesn’t perform data retrieval and storage in the middle of an equation).

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

Note:  
After 20 years of putting in rule engines that can also be used by non-developers to add, modify, or remove logic to a system, I’m still relatively against them. They’re a relic from an era where deployments were nontrivial and not done often, which frustrated business users. But Non-developers do not usually have the same instincts, experience and tooling to develop automated user tests around their logic and therefore the cost of change is much higher, due to manual testing, insecure due to access to production data, and riskier of affecting end users by introducing incorrect the non-code rule engines are updated. I remain slightly more comfortable with rule engines being integrated that only developers have access to, because they have more of a habit wrapping their updates with static and dynamic tests – but I still question whether learning a Domain Specific Language (DSL) to process logic, rather than a programming language, adds real value.

#### Workflow Management Services

Workflow management service integrate with notification services to allow multiple roles to contribute to a sequence of steps performed asynchronously. Examples might include inviting and approving users to roles, approving large value payments, etc.

Workflow management is more complex than it looks and is best served by offloading the work to a dedicated service.

The credentials required for informing and/or querying this service are of course retrieved at startup from the credential store mentioned earlier.

### Enterprise Services

Services don't work in a vacuum. There are often enterprise services that are expected to be hooked up to the service, to delegate to:

* **Shared Drives:** although far from ideal, many organisations still share networked drives in order make documents available to peers.
* **EDMS:** Electronic Data Management Services (e.g.: SharePoint, FileNet, etc.) are common to more mature companies -- once they've moved past Share Drives.
* **Enterprise Services:** other common enterprise services that may need hooking up include Financial, Personnel, Training, Case, Bid, CRM, Contract, Risk, Configuration services.

## Final considerations

### Sub Networking

This paper focused on describing common dependency services for systems, custom or not, and will only touch lightly on a related aspect: deployment of the infrastructure to support these services.

It’s best practice that devices and services are isolated from each other via subnets protected by traffic rules that only accept incoming traffic from known sources. For example, its expected that database devices are within a subnet that only accepts direct traffic on specific ports from the webservers in another subset.

The channels must be secured by encryption where possible.

Finally, monitoring can be established on these channels, collecting information for reporting and/or alerting subscribed maintenance and security stakeholders.

### Outward Data Migrations

#### Data Backups

Data backups have to be taken regularly to provide a means to restore system in the event of critical system failures.

Backing up relational data is a straightforward process as the feature is part of most Relational Database management systems (RDMS). It just needs to be initiated at a regular cadence (e.g.: full backups every night, incremental backups every 5 to 15 minutes).

Backing up other datastores can be a little more complex to setup, as the datastores may not have a backup feature in their own right. In such cases, the data has to be duplicated to another data store – although these duplications are often already a feature of the cloud provider providing the datastores.

#### Reportable Data

Exporting reportable data to a separate reporting database has already been discussed earlier under Reportability above.

#### Data warehouses

Data is exported to external data warehouses either from the transactional database or from the reporting database, depending on the purpose of sharing data with a data warehouse.

The exporting process can vary drastically depending on what’s available. The preferred method is to deliver information to APIs endpoints the data warehouse provides, or may rely on an older approach, such as creating xml or csv files delivered to a shared drive that the data warehouse will pick up later.

### Inward Data Migrations

#### Backup Restorations

The restoration of relational databases is also relatively straight forward as the capabilities are offered by most RDMS.

In the case of other datastores, restoring a datastore is often done by switching pathing to a duplicated datastore.

## Conclusion

Even a high-level understanding of what dependency and integration services are common to business services and what they are for, and what they enable, permits better planning of the sequence of work items by delivery stakeholders, requiring less retrofitting of cross-cutting and essential services that is not only costlier to do later, but with less certainty of success.

Appendices

Appendix A - Document Information

### Author & Collaborators

Author: Sky Sigal, Solution Architect

### Versions

* 1. Initial Draft
  2. Minor updates

### Images

[Figure 1: Outgoing Integrations to Dependency Services 8](#_Toc151367525)

### Tables

**No table of figures entries found.**

### References

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

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
| Sandy Britain, Enterprise Architect |  |
| Amy Orr, Data Domain Architect |  |
| Russell Campbell, Project Manager |  |
| Rodney Snell, Business and Technical Lead |  |

### 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’s Glossary.

##### IT

: acronym for Information, using Technology to automate and facilitate its management.

##### ICT

: acronym for Information & Communication Technology, the domain of defining Information elements and using technology to automate their communication between entities. IT is a subset of ICT.

##### RDMS

: acronym for Relational Database Management System. A service to create, manage databases and manage queries to them.

##### SSO

: acronym for Single Sign On. A way for IdPs to create authentication tokens that permit users to authenticate once and provide in return a token that will be granted entrance by multiple services.

1. If it was done before, presumably the service would be bought or subscribed to rather than built. [↑](#footnote-ref-2)
2. Best practice is every 3 months, max. [↑](#footnote-ref-3)
3. See Malware Detection Service, described later in this document. [↑](#footnote-ref-4)
4. They’re expensive, to purchase and develop on, and maintain. So it’s best that services are completely self-reliant without them, and the API Gateway just reroutes traffic as needed. [↑](#footnote-ref-5)
5. For example, linking to the user’s current tenancy without the reliance on a dns subdomain or url variable to direct the server. [↑](#footnote-ref-6)
6. So much so that it swanned a whole era of incorrect system design that referred to a “data layer”, at the expense of mentioning other necessary integrations. [↑](#footnote-ref-7)
7. *when configured and used correctly.* [↑](#footnote-ref-8)
8. This becomes important when later developing Subnets around both web servers and storage services. [↑](#footnote-ref-9)
9. Relational databases are not especially suited to meeting streaming needs, and it increases the storage space and time needed by backups, increasing running costs. [↑](#footnote-ref-10)
10. Can be considered either a specialised type of in-memory data store, or a unique service type in its own right. Doesn’t really matter. [↑](#footnote-ref-11)
11. This requires high responsivity of APIs. [↑](#footnote-ref-12)
12. Nato embargoed, etc. [↑](#footnote-ref-13)