Support Services
Directorate

KENT POLICE ESSEX POLICE

Supporting policing
in Kent and Essex

TDA Review Document

High Level Lite Design:
Data Transformation

Proposal Summary

Project Name	Data Transformation	**Project ID**	KEP0018
Project Description	This project seeks to centralise and modernise the existing data reporting landscape and toolset whilst migrating the Management Information and Business Intelligence reporting from the on-premises Business Objects to Power BI, in Microsoft Azure. Business Objects will be retired as it is now a legacy capability and newer reporting tools can better serve data to the users. Core to this project is seeking to drastically reduce the circa 53,000 reports that exist in Business Objects, and re-engineering the way data is pulled from source databases and data systems, into an environment suited specifically for analysis and reporting, whilst retaining the business logic already written in Business Objects. Using Microsoft Azure cloud and its Software as a Service (SaaS) product Fabric to automate the business systems' data into a common data lake platform will make user reporting, data engineering and data science more accessible and easier to govern. This document was initially published, as a first version, to inform TDA what was planned for the data transformation project. Having introduced Fabric and the medallion architecture, and as sandboxed development of some of the core components have been readied for the new environment. The updates in this version		

	have provided connection information as well as for the service components e.g. DevOps and Fabric Workspace management, to provide information about the workings in the new environment. In subsequent updates for further review there will be detailing of Copilot and iBase as they are investigated and standards for their use adopted, along with the incorporation of new geo-spatial capabilities in Power BI's mapping visuals to compliment intelligence requirements. These will be presented for further review that will enhance what is presented thus far and anticipated as a ready-state for sign-off of the presented version. This HLD is intentionally high level – it seeks to introduce Microsoft Fabric as the replacement for Business Objects and describe the key components that will be introduced and how they will be implemented. Accepting this design as it stands allows the Project to begin to deliver the new architecture and Phase 2 (described below) – with detailed updates to this design being produced throughout the lifecycle of the project for subsequent approval by TDA, alongside any additional implementation documentation.
Technical Proposal Owner	Author: Tarek Moghul - Data Architect - Contractor
Total Cost Estimate	£1,460,750 - (the budget for 3 years of project) Licence costs - £343,095 Contractor costs - £893,681 Training - £103,926 Miscellaneous costs - £120,048
Options	
1. Accept this HLD **2. Reject this HLD** **Please note which HLD Version and Option number you are signing for in Date column**	
Recommendation	
It is recommended the TDA accept option 1	

Document control

Version History			
Version no.	Version date	Author	Summary of change(s)
0.1	27/01/2025	TM	Create document
0.2	10/02/2025	TM/GP	Multiple updates following initial review, also incorporating information gathered from Discovery work undertaken by the project to help inform the design/delivery
0.3	25/02/2025	TM/GP	Version issued to BW and ON for initial internal review
0.4	11/03/2025	TM/GP/BW	Further updates following initial pre-TDA review
1.0	19/03/2025	GP	Removal of final comments – final minor updates – version released to TDA for review.
1.1	07/05/2025	TM	Updates to document from Review Tracker comments
1.2	14/07/2025	TM	Update to 1. Project Timeline Addition of Fabric to On-Prem VMs for data connectivity Sections: 1, 2.1.2, 2.1.5, 2.3.1, 2.4.1, 2.4.2 2.1.2 Target Architecture (Azure) updates referring to Azure and Fabric Data Factories, aligning VM (SHIR / Gateway) diagrams, platinum layer for enriched data 2.1.4 Fabric Workspaces 2.1.5 DevOps and Continuous Integration/Continuous Delivery 2.2.1 Backup and Data Resilience, RPO / RTO 2.2.2 Backup Schedule / Periods 2.4 Technology Architecture - DevOps Jump Box config Design Decisions: DD06, DD07 - Anonymised dev data and SHIRs for VMs 5. Added VMs and Purview installation & config to Implementation Tasks Appendix 8.5 Naming Conventions

Sign-off details (IT SMT)				
Sign-off authorities	Role	Date, Version & Option	Signature	Comment
Fiona Brown [FB]	Chief Information Officer		Review not requested	
Phil Bartholomew [PB]	Head of IT SACS			
Alex Allen [AA]	Head of IT Solution Delivery			

Sign-off details (IT SMT)				
Sign-off authorities	Role	Date, Version & Option	Signature	Comment
Matthew Chinavicharana-Mole [MC]	Head of IT Portfolio & Business Engagement			
Steph Gill [SG]	Head of IT Service Delivery			

Reviewers (Core TDA)				
Name	Role	Date Version & Option	Signature	Comment
Mark Thomsett [MT]	End User Computer Development Manager			
Gavin Purnell [GP]	Technical Solutions Architect			
Paul Saunders [PS]	Server Infrastructure Manager			
Terry Warby [TW]	Network Services Manager			
Vita Steward [VS]	Information Security & Business Assurance Manager (Kent)	25/04/25 V 1.0		Comments added to tracker for future consideration
James Wyatt [JW]	Information Security Officer (Essex)			
John Knowles [JK]	Commercial Applications and Database Manager			I am not comfortable to sign off until the backup detail of RPO and RTO is defined
Reviewers (Additional)				
Nicola Endacott	Head of Intelligence and Data Analysis - Kent			
Dr Natalie Mann	Head of Research and Analysis – Essex			

Contributors		
Name	Role	Contribution
Lewis Blackford	Software Dev Team Leader	Business Objects data re-engineering to Azure / Fabric
Kishore Kalla	Data Engineer	Metadata Framework, DevOps, VMs, SHIRs / Data Gateways, Weeding

Contributors		
Name	Role	Contribution
Rexford Osei-Aboagye	Data Engineer	A4E source data and ETL, data curation and presentation, Data Warehouse design, Weeding
Gavin Purnell	Technical Solution Architect	Content, structure and document flow
Debbie Grant	Project Manager	Design, Implementation and Risk
Ben Wallis	Technical Solutions Designer	Peer review

Contents

Table of Figures

1. Project Timeline

Stage	Timescale	Key activities
Discovery	September 2024 to March 2025	• Review of existing data and reporting landscape, and produce high level design brief with proposals for the solution • High-level map of systems that will feed the transformed data environment and reports, showing the consolidated reports, they will feed (Design Decisions - DD01) • Discovery exercises that can demonstrate some key functionality of the proposed system - Trial licence of Fabric to test features within the Fabric interface - Produce and send MoJ data extract using Power Automate - Existing (BOb) real-time STORM dashboard in Power BI - Risk Register Bob report recreated in Power BI - Rebuild A4E source views in Oracle Local Copy - Testing Data Science capabilities in Fabric - Metadata Framework functionality - coding the data flow Design Decisions - DD04 • Business use cases • High-Level Design Document produced and approved • Decoding the SAP environments in Business Objects to understand the scale and complexity of what needs landing in the new environment
Stage 1 Platform Development	March 2025 to December 2025	• Setting up the environments for Sandbox, Pre-Prod, Prod • Setting up DevOps - code repositories and Project Management boards • Creation of Access Control and Governance groups, and tagging • Prioritisation of business use cases • Setting up Azure Landing zones • Setting up connectivity VMs for data access • Carry out penetration testing on new environment and remediate whatever identified • Work phased as source to destination report migration of Analytics for Everyone (A4E), for SAP and STORM data, to build scalable process for subsequent data sources. This will allow earliest deployment to production environment and pen testing • Development of data engineering to replace Business Objects processes • Migration of 5 data source reports to new environment and redeploying of associated Power BI reports as specific tranche. Will give source to destination process and live deployment that can be repeated for each additional source system • Incorporate data sources in new medallion flow • Migration of Business Objects calculated fields to new semantic models • Recreate priority Business Objects Universe as new-world semantic model - Possibly Harm model • Delivery of Stage 1 Functional Requirements
Stage 2 to 4	January 2025 to March 2026	• Decommissioning of existing underlying infrastructure of A4E and archival of A4E's data

Data Source Transition		• Source by source inclusion to the medallion flow – Continuous Integration/Continuous Delivery (CI/CD) type approach to allow for gentler release schedule than big bang • Phase release cycles to logically clustered user profiles based on role • Design, document, sign off and test provision of data/reports to external third parties - if / when required. • Management Reporting - Consolidate billing information with platform and transformation pipeline execution metrics in Log Analytics workspace. Create monitoring reports and dashboards for real-time and historic operational metrics • Delivery of Stage 2, 3 and 4 Functional Requirements, including Co-Pilot, Geo-Mapping and data sharing with external partners

2. Target (To-Be) Technical Architecture

2.1. Overview

The current data transformation process uses SAP Data Services to curate source data and store it in Oracle Data Marts that are served to the users via Business Objects Universes. In essence the transformation and preparation of data currently in Business Objects will be carried forward to Microsoft Azure and re-engineered using the Azure SaaS toolset - Fabric - within the KEP Azure tenant.

The target architecture divides the transformation load across Raw (Bronze) / Curation (Silver) / Warehoused (Gold) layers - the medallion architecture - in a data lake within Microsoft Fabric that provides easier processing, archiving and management of the data in its various stages as well as the application of security identifiers, mastering / cleansing of the data, governance and audit (Design Decisions - DD03).

One of the primary design considerations of the new architecture is to decouple the extraction of the data from the source systems / databases with Fabric, from all the downstream processing and uses of that data. By only querying the source data when necessary (i.e. for the relevant refresh interval for that data), we ensure we minimise the impact to those systems. Once the data is in Fabric we can take advantage of existing capabilities within Fabric, such as data engineering, transformation, reporting, data science etc - as well as new capabilities as and when they become available, and the business need is confirmed. The use of Fabric will also allow the Analysts teams within the business a greater opportunity to share and leverage the capabilities of Fabric, which within the current Business Objects environment can only be used by the Data Engineering Team within IT Services.

As part of this design, we will re-architect the Analytics for Everyone (A4E) system that is in the KEP tenant to be an initial deliverable from the new Fabric environment. We will migrate from the use of SQL databases and Azure VM to the new Fabric environment (see this section for further details).

The Legacy Digital Data Store (LDDS) is in the KEP tenant, amalgamated with our Tenant's Identity Subscription. Ideally in the long term this would be in its own dedicated subscription - but this is outside the scope of this project.

2.1.1. Existing Architecture (Business Objects)

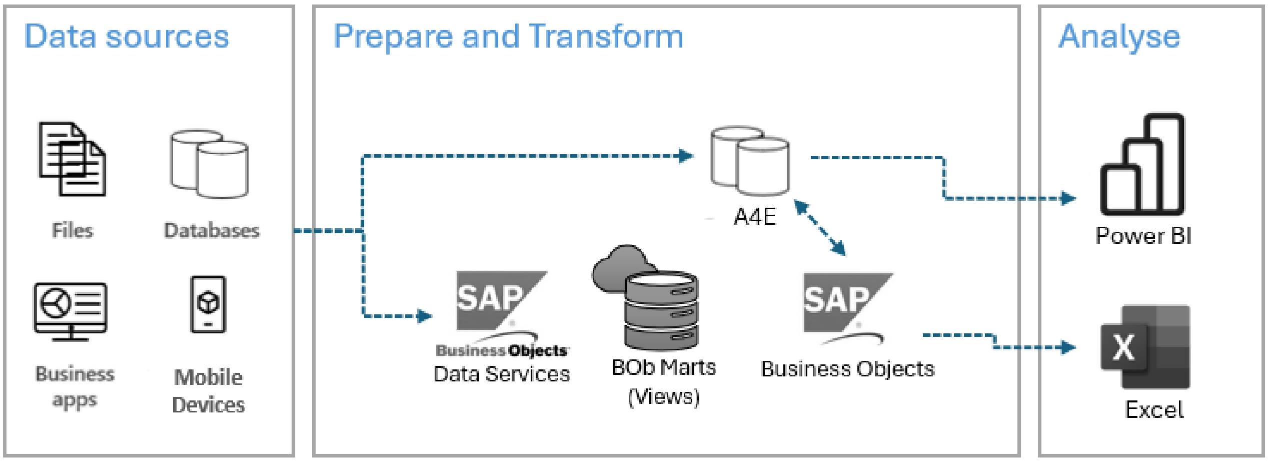


Figure 1 - Existing Architecture

Data from source systems is currently pulled into SAP Data Services for transformation and application of custom calculations and then made accessible to users via Business Objects Universes. A4E separately processes the data from Athena, Bail, STORM and SAP whilst also taking in transformed data from Business Objects for curation. A4E currently serves its report outputs to Power BI. Most of the current reporting is through Business Objects and delivered to customers via the BI Launchpad interface or email, which if required can be exported to CSV files or Excel. A4E serves Essex users only and Business Objects serves both Kent and Essex.

2.1.2. Target Architecture (Azure)

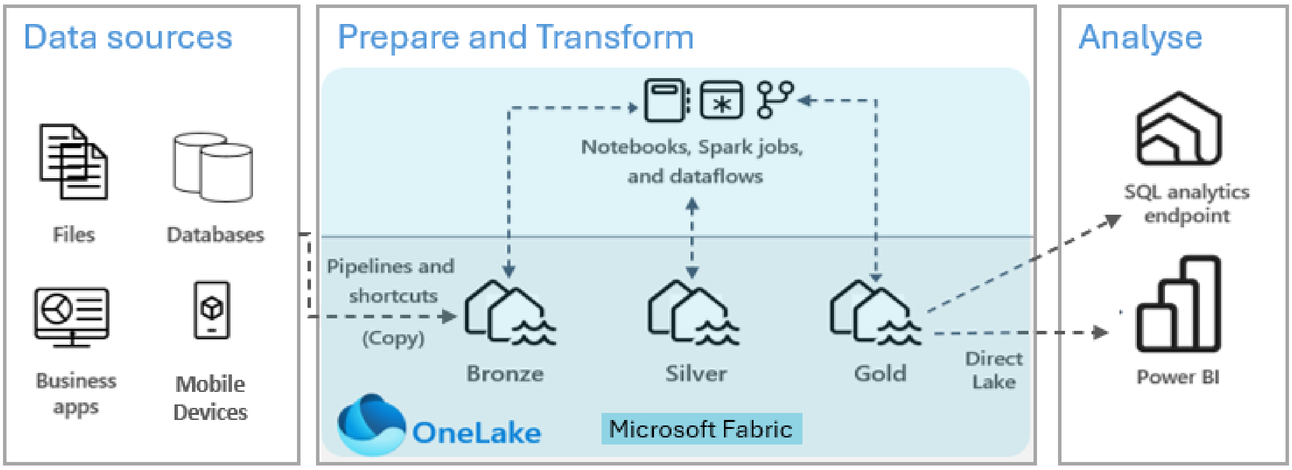


Figure 2 - To-Be Architecture

Source data is pulled into the bronze layer of Fabric's data lake (OneLake), using Fabric and Azure Data Factory pipelines, to land all the required data for transformation that will be consumable via Power BI. This will form a raw / unaltered copy of the source systems' data that is required for transformation, for reporting and will constrain the number of read hits on the source systems to just those required when ingesting data rather than when output reports are opened and / or refreshed (Design Decisions - DD03). The data engineering within Fabric will comprise of scripted tasks using Spark to take advantage of its highly efficient and fast processing capabilities. These will be carried out in batches where practical, collated into logical code notebooks to automate the creation and running of the data factory's pipelines. Notebooks are also a neat method for storing code scripts that can allow sections of the script to be tested without having to run the entire script. This notebook / script operation is the equivalent to Business Objects Data Services operation. The execution of scripts will be automated through a Metadata Framework (see below - Section 2.1.8 and Design Decisions - DD04) as a controller to the ingestion and transformation tasks.

Data is cleaned and converted in the Bronze to Silver layer transformation process to standardise data types, in the Silver layer, from the various flavours of data and their respective system's types. The data held in the silver layer is only what has been brought in on the latest source data refresh.

The Silver data is then checked against what already is held in the gold layer's warehouses, in the Silver to Gold transformation process. The data is broken up into calculatable / aggregate-able (predominantly numeric) Fact tables and descriptive (text) Dimension tables, which are assigned key (ID) identifiers that tie in with the Gold layer's warehoused data. This is an approximate equivalency to the existing Business Objects BOb data marts. Logical grouping of the warehoused data, equivalent to the existing universes in Business Objects, are created on the gold layer to which Power BI reports will be able to connect directly. These are the data models that Power BI uses that are structured exactly as an individual Power BI file's data model, known as Semantic Models. These exist to link directly to Power BI's reports, without the need to connect to any of the preceding data steps in silver, bronze layers or source databases, eliminating the need for the source data to be re-queried every time a report is run. The semantic models will share common datasets and calculations across multiple reports that will be grouped logically for access and users (See Design Decisions - DD01).

The way the data is made available for reporting in Power BI, utilising the medallion architecture, de-couples the reports from the source systems, their data and the associated transformation process required to make this data presentation ready. The additional load that current reporting queries place on the source systems is moved away to a reporting copy of the data. The copy allows for greater flexibility in the transformation operations and is effectively a cached, backup copy of the same data that does not require re-reading from the source every time a report is opened and / or refreshed and is available in the event of disaster recovery. The data is Extracted Transformed and Loaded, following an ETL pattern, as shown in Figure 3 below. The associated transaction costs of source data re-reads are greatly reduced to a scheduled one-time operation that the subsequent medallion flow tasks utilise.

The addition of a segregated area for creating bespoke semantic models is shown in the gold area of Figure 3 below, as an enrichment area. In keeping with the precious metal nomenclature of the medallion architecture it is referred to as platinum. Whilst this is euphemistic in nature and not a physical layer that requires additional processing, it is a consolidation area where the data models that will have custom calculations and measures added to them are logically grouped. These measures / calculations would be equivalent to Business Objects variables, that will give specific reports or their collective grouping relevant additional data fields to report on. These would typically be summary measures that update dynamically on reports when filters and slicers are applied by the user.

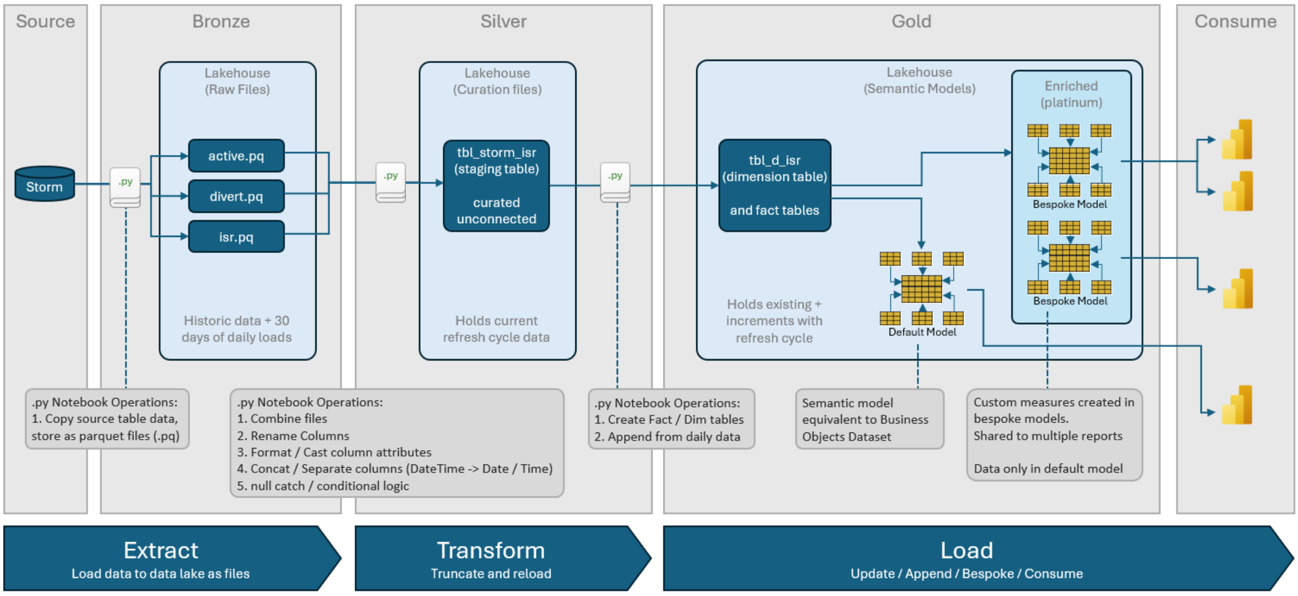

Source
Bronze
Silver
Gold
Consume
Storm
.py
Lakehouse
(Raw Files)
active.pq
divert.pq
isr.pq
Historic data + 30 days of daily loads
.py
Lakehouse
(Curation files)
tbl_storm_isr
(staging table)
curated
unconnected
Holds current refresh cycle data
.py
Lakehouse
(Semantic Models)
tbl_d_isr
(dimension table)
and fact tables
Holds existing + increments with refresh cycle
Default Model
Enriched
(platinum)
Bespoke Model
Bespoke Model
.py Notebook Operations:
1. Copy source table data, store as parquet files (.pq)
.py Notebook Operations:
1. Combine files
2. Rename Columns
3. Format / Cast column attributes
4. Concat / Separate columns (DateTime -> Date / Time)
5. null catch / conditional logic
.py Notebook Operations:
1. Create Fact / Dim tables
2. Append from daily data
Semantic model equivalent to Business Objects Dataset
Custom measures created in bespoke models.
Shared to multiple reports
Data only in default model
Extract
Load data to data lake as files
Transform
Truncate and reload
Load
Update / Append / Bespoke / Consume

Figure 3 - Medallion ETL Data Load Pattern

Fabric has native tooling for Data Science, additionally to the data engineering and other Azure resources that would otherwise require subscriptions (at additional cost) and components, that will be used to build, maintain and update the data transformation tasks. Data scientists will be able to use Fabric's Integrated Development Environment (IDE) without having to rely upon third party tools that may not pass InfoSec's scrutiny.

Data from both the bronze and gold layers will be accessible for data science work. Any additional data required, that is not drawn from the source systems, can be uploaded to the Bronze layer as raw data.

2.1.3. Azure Subscriptions and Environments

The subscription hierarchy within Azure will be Pre-Production and Production environments, in accordance with Microsoft's guidance and PDS Landing Zone specification, as separate subscriptions. A sandbox subscription exists that will be pay-as-you-go enabled, as a secure data enclave for testing outside of the working environments. The sandbox will be available for testing features and general experimentation outside of the data transformation environments in order not to clutter any structured development and testing areas. The sandbox subscription is outside of the scope of this document and only referenced as it is part of the wider Enterprise Landing Zone architecture.

The subscription tiers for these environments are recommended to use the Premium capacity tier of Fabric for the Production environment (F64) and the lower cost (F4) tiers for Pre-Prod (Development). The live production environment must include the capability to share data and reports with external partners and stakeholders, which is only available in the Premium capacity tiers of Fabric - F64 and above. These premium capacity tiers are a Fabric-equivalent of the existing Power BI Premium subscription used by A4E that have been packaged into one SaaS product. A noteworthy point here is the governance of any data that is shared with external parties will be managed by Purview alongside the access control and security with Active Directory / Entra. The Pre-Prod environment doesn't need to share any data externally and is specifically for carrying out development. The Prod subscription will additionally hold an environment for User Acceptance Testing (UAT) for live data testing by the business users as well as for training.

The pre-prod environment will separately (from live) hold an extract of the live data that is constrained to a smaller number of records (approximately 100k rows per data source) than the actual live data. This 'dev' subset of data will be identical in structure to the live data and will also have all data in fields that contain personally identifiable information masked / obfuscated. This allows for a lower capacity tier of Fabric for the development environment that in turn will process fewer records on a longer refresh cycle to that in prod. This is primarily to reduce operational / transactional costs. This test / dev data has been investigated separately and detailed in a Testing Strategy document, referenced in Appendix 8.4

Figure 4 below is a conceptual diagram, based on Microsoft's best practice Opinionated Architecture, showing how the Fabric environment will fit into the existing KEP Azure tenant. This design aligns with the PDS standard and allows us to strengthen the overall Azure environment to a robust, enterprise-grade level. Furthermore, with the policies and access controls that are a native feature of this design we have the governance fundamentals built in from the beginning. Whilst this architecture has the various components embedded across the subscriptions, it is ideally suited to incorporate future additions and enhancements with a modular approach that compliments the overall established structure.

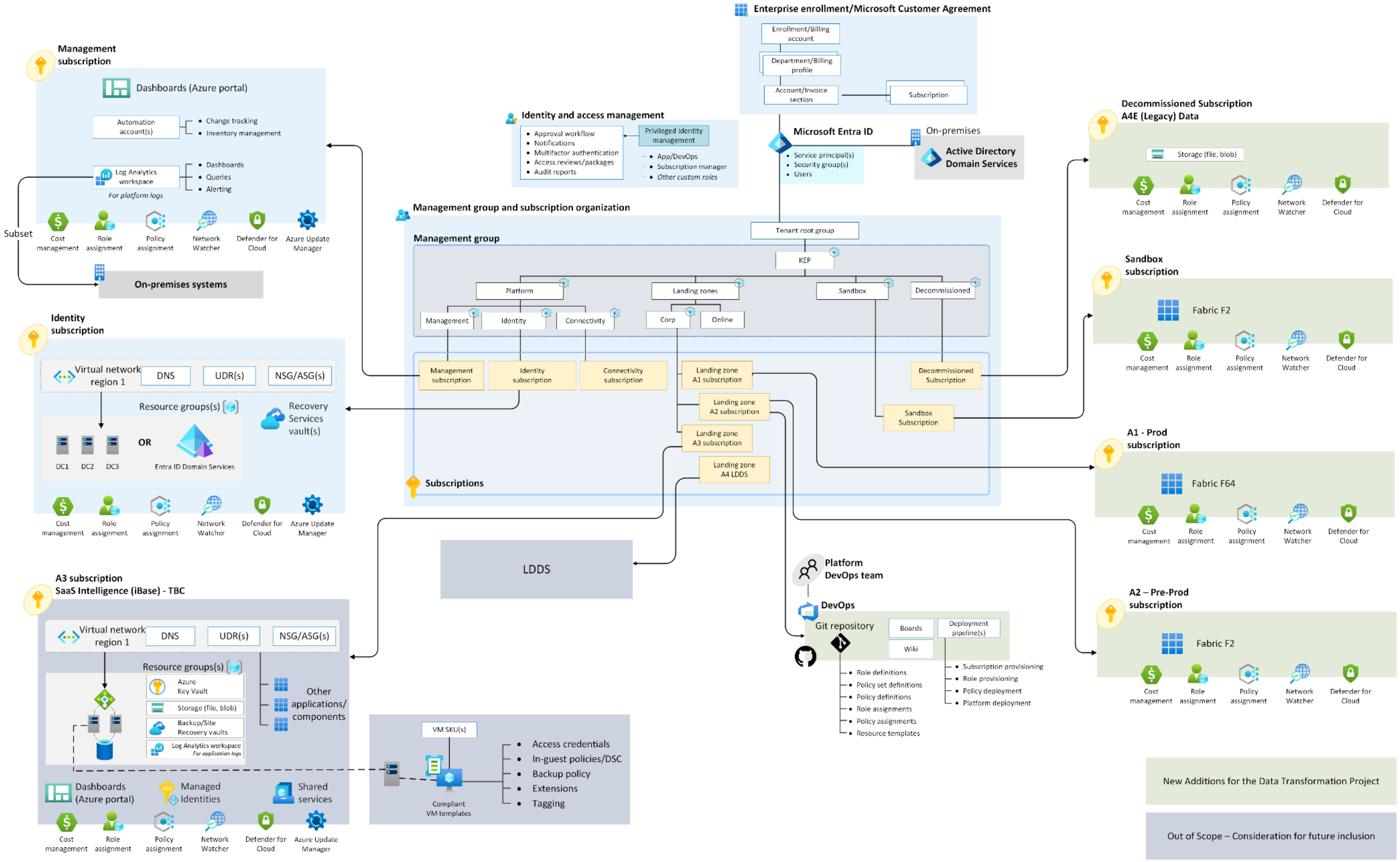


Figure 4 - Azure and Fabric Subscription Landing Zone Architecture

2.1.4. Fabric Workspaces

Beneath the Pre-Prod and Prod subscription's hierarchy, with their respective Fabric resources, are the operational workspaces that will hold all the data, pipelines and reports. These workspaces are a means to ensure good housekeeping of all that is added to Fabric's data lake.

Segregation of the workspaces is designed to contain data from each source that is ingested, separated from each other. These are replicated across the medallion layers where they are required for deployment of pipelines / reports etc. by DevOps deployment pipelines. Additional workspaces for operational purposes such as data engineering and data science that are required outside of the medallion flow are similarly available.

The interaction between working domains and medallion layers is represented in Figure 5 below.

For each new data source added to the ingest flow and process, a new workspace will need creating. There is no limit to the number of workspaces that can be created, however there is a 1000 item limit to the contents. For data sources that would breach the item limit consideration would be needed as to the best way to distribute these across multiple workspaces. These might follow Data Source 1, Data Source 2 etc. or Data Source Logic Group (HR / Finance etc.). Each case will need to be decided on demand.

As the data travels through the medallion layers, the destination lakehouses and their semantic models to which Power BI reports will connect are handed on (replace) to the data and processing centric warehouses. Access to these will be governed by the standard Azure AD / Entra permissions and will typically be segregated by Data Engineering, Data Science and BI Engineering (Analyst) Teams.

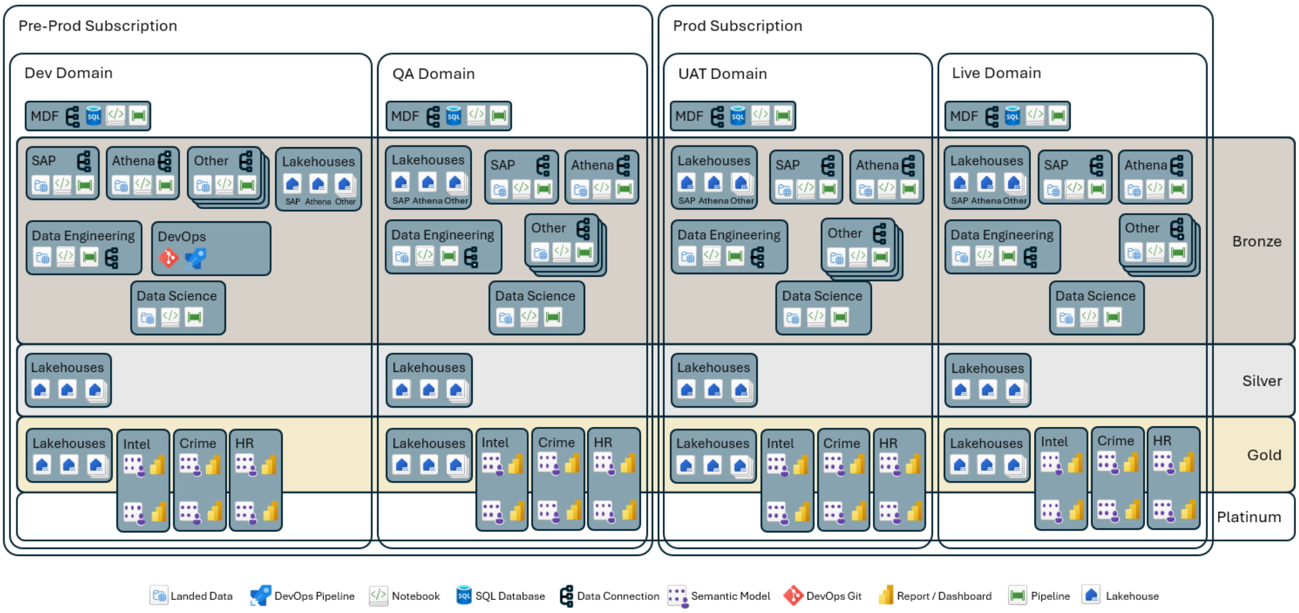


Figure 5 - Fabric Workspaces

2.1.5. DevOps and Continuous Integration/Continuous Delivery

The project will use Azure DevOps to manage code repositories and deployments across environments, which also double up as operational version control and backup. This is an industry standard management tool within Azure that utilises best practices from the development world. Incorporating all the process steps from task allocation and management through single source code storage for onward deployment.

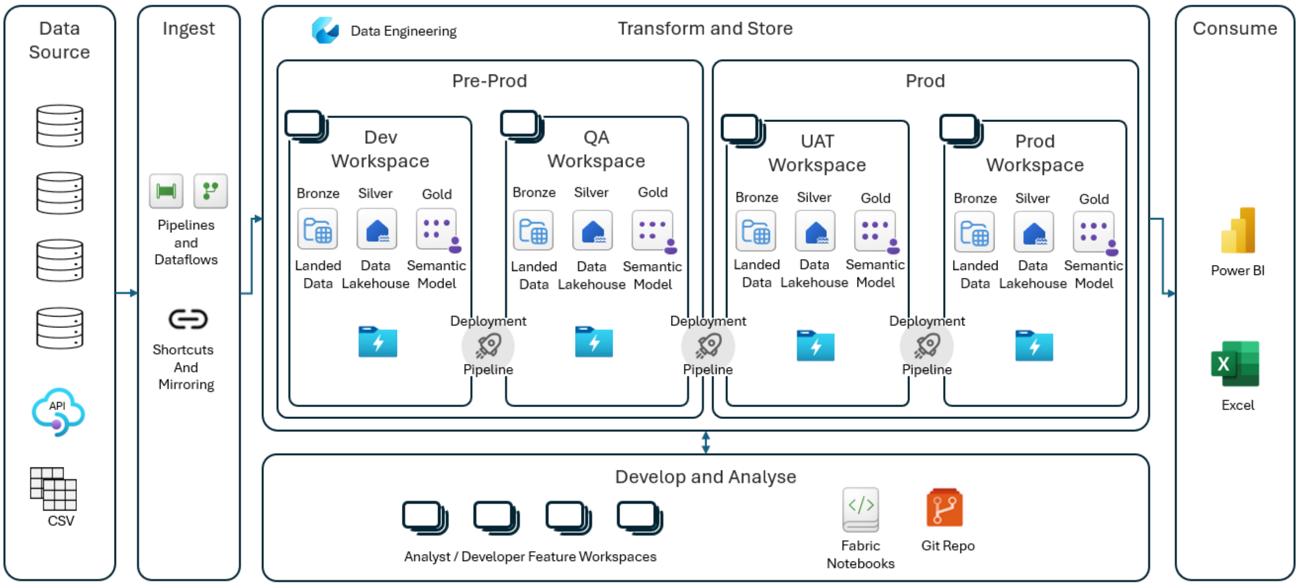
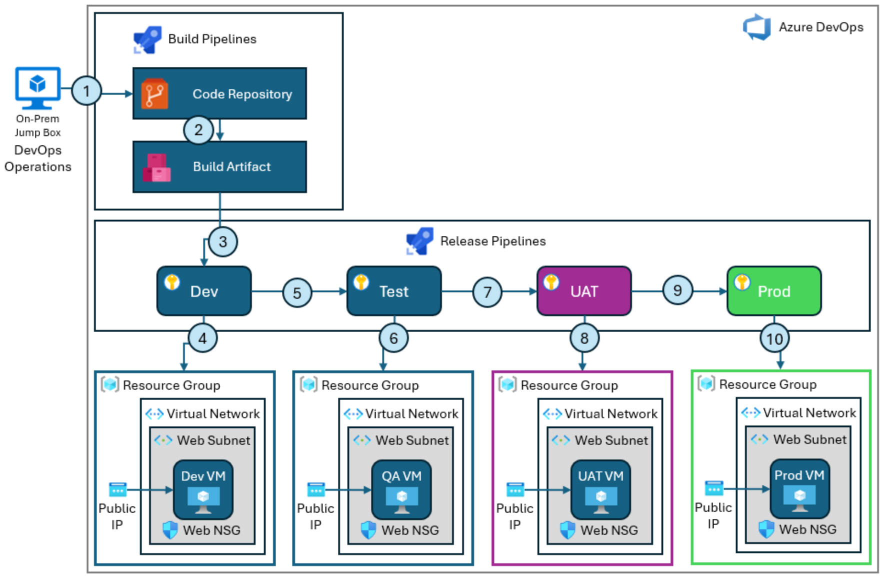


Figure 6 - CI/CD Deployment Pathway

As shown in Figure 6 above data engineers, analysts and data scientists will work in a development workspace before promoting their changes into the QA workspace to validate what has been built. This allows for parallel working, with DevOps managing the merging of work from shared tasks across environments. Once the changes are ready for business scrutiny these will be deployed into the UAT workspace, before finally being deployed to the Prod (live) workspace, once approved by the business users / testers. This will be managed using DevOps deployment pipelines using DevOps native YAML and Bicep scripting that pushes code updates across environments and replaces connections from dev data to live data. See Figure 7 below showing the sequence and deployment process.

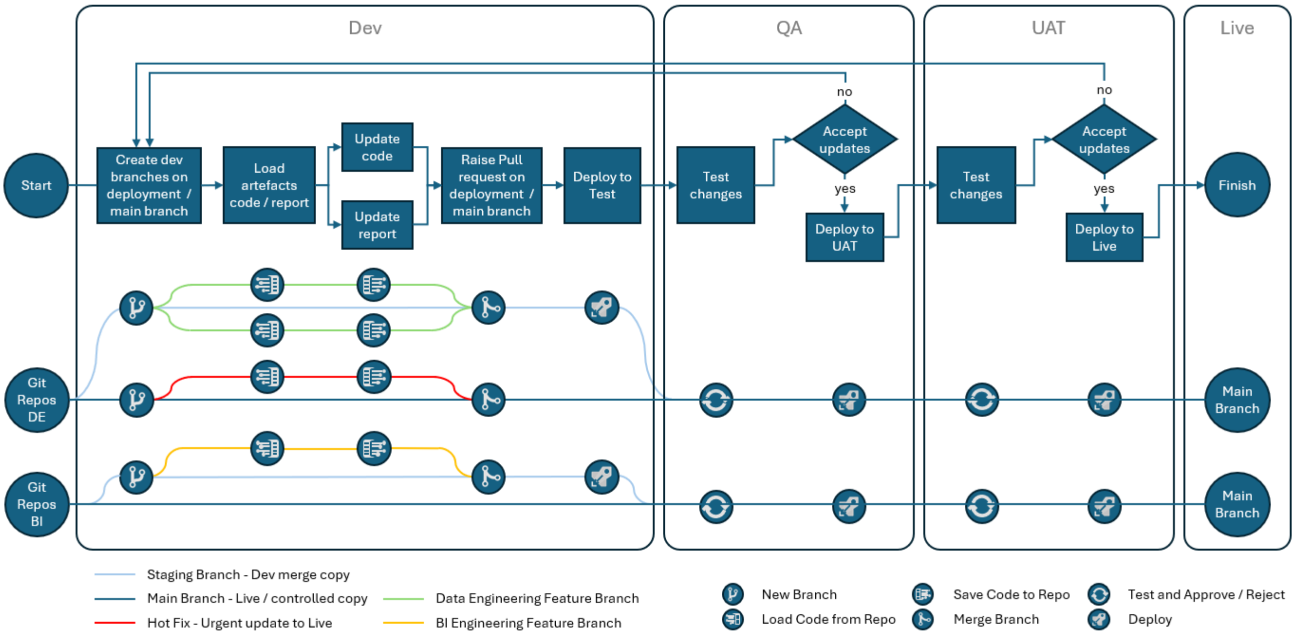


1. Code Development - Developers create feature branches, implement changes, commit, and push code, via Jump Box where VS Code is installed
2. Merge & Build - Code is peer-reviewed, merged into main/master, and the pipeline prepares build artifacts (packages, Fabric items, SQL scripts, etc.)
3. Deploy to Dev - Build artifacts are automatically deployed to the Dev environment
4. Validate in Dev: Deployed to Dev and developers perform unit tests / smoke tests in Dev to ensure functionality
5. Deploy to Test - If Dev validation passes, code is promoted and deployed to the Test environment
6. Verify in Test - The test team (or automated regression tests) validate the solution against requirements
7. Deploy to UAT - Once Test verification succeeds code is promoted and deployed to the UAT environment
8. Validate in UAT - Business users (UAT team) validate functionality against requirements and sign off
9. Deploy to Live (Prod) - After UAT acceptance, the release is promoted and deployed to the Live environment
10. Post-Deployment Verification - Smoke tests, monitoring, and validation are performed in Live to confirm stability

Figure 7 - DevOps Deployment Process Sequence

Access across the Pre-Prod environments will be based on use case requirements, for data engineering - curating and making the data available, and for creating the Power BI reports. General users of Power BI will be served their reports and dashboards from the live Production environment. These might be considered by their Personas - Data Engineer, BI Engineer, Data Scientist, User etc, requiring various degrees of granular access to the data across the medallion layers and workspaces.

The diagram (Figure 8 below) shows the process flow of creating new and updating existing data engineering and BI reports. Moving these through the medallion layers to live / published where all can be consumed. DevOps branches (shown in the lower part of Figure 8) demonstrate how development and deployment branches are managed for the deployment across the environments. The process allows for simultaneous work to create and / or update the data engineering scripts and processes as well as work on the Power BI reports. No need to open a file that is locked to one user whilst it is updated, nor fear of updates being overwritten by changes made after.



Dev
QA
UAT
Live
Start
Create dev branches on deployment / main branch
Load artefacts code / report
Update code
Update report
Raise Pull request on deployment / main branch
Deploy to Test
Test changes
no
Accept updates
yes
Deploy to UAT
Test changes
no
Accept updates
yes
Deploy to Live
Finish
Git Repos DE
Git Repos BI
Main Branch
Main Branch
Staging Branch - Dev merge copy
Main Branch - Live / controlled copy
Hot Fix - Urgent update to Live
Data Engineering Feature Branch
BI Engineering Feature Branch
New Branch
Load Code from Repo
Save Code to Repo
Merge Branch
Test and Approve / Reject
Deploy

Figure 8 - Development Flow - Process and DevOps CI/CD

A Git code repository in DevOps holds the master copies of the developed code and reports, which is the source from where updates are carried out and deployed. Version control is maintained this way as is the merging of parallel development streams by the developers.

The master / controlled code and reports (artifacts) belong to the main branch. When creating new artefacts or updating existing ones, a working (staging) branch is created, which exists in parallel with the main branch. Figure 9 below shows Azure Boards at the start of the development task, where tasks are identified and then how these interact with the steps involved in creating and deploying artifacts. Associated to a task, each developer (data engineer / BI engineer / scientist) then pulls the artifact they need to work on from the staging branch into their own working branch. These are named with the same task identifier in the project management list, in the boards. Once updates have been carried out, they are merged with the staging branch ready for deployment to the next environment along. If parallel development is carried out, then deployment can be held until all development branches are ready to be merged and changes consolidated and accepted. DevOps manages the merging process allowing the developers to select which changes to accept.

Adopting this approach to version control and artefact-driven deployment ensures that control is maintained centrally over the code and reports provided to the business. Updates are all managed by the data engineering and BI engineering teams from a central point. It is a disciplined approach to development that follows best-practice principles and aligns neatly with access control protocols, referenced in 2.1.6 below. When using artefacts from the repository, all updates to existing reports and code start from the left (Figure 8 above). This means any updates to existing reports that might be carried out locally would become uncontrolled copies and be lost when subsequent updates are deployed via the DevOps route. This, when coupled with controlled access to development resources and environments, would eliminate the scenario where tweaked copies of tweaked copies of reports accumulate in the report list. See Design Decisions DD08.

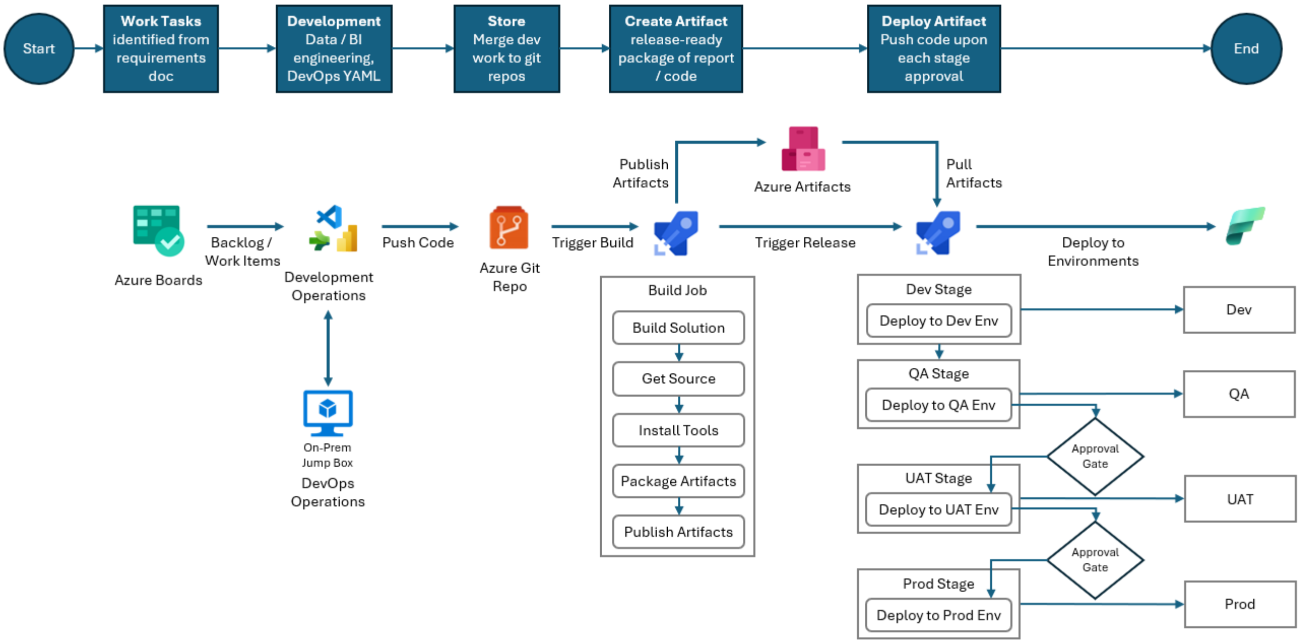


Figure 9 - Deployment Flow from Task to Live

The process is known as Continuous Integration / Continuous Deployment (CI/CD) that allows new reports to be created, and existing ones updated whilst they are being tested, approved and submitted to the users. This will also keep data engineering code and reports controlled correctly by utilising DevOps and Git for version and source control, as opposed to 'save-as' copies.

The Azure Boards (shown above in Figure 9) are the springboard from which the DevOps process runs. Once a task is identified it is listed in the board and assigned an ID. This allows for effective project management, with the task being updateable for status and notes. Utilising kanban board view allows for collective viewing in scrums to quickly see and update the status of each team member's tasks. Importantly for the code deployment, by attaching the task's ID to code branches the segregation between team members' code is maintained. History of tasks are maintained that can also be utilised for management reporting as well as retrieval of actions carried out on a task for back testing / investigating what was changed during the development cycle.

2.1.6. Access Management Hierarchy

One of the key advantages of using the Microsoft technology stack in this design is the ability to easily re-use established components such as Azure Active Directory and Entra to govern access and management groups, often with the ability to re-use standard roles and existing role-based access groups. The hierarchy is shown below in Figure 10. Azure Active Directory and Entra will be used throughout to govern access and permissions - from the end user access to data, to the data engineers, data scientists, analysts, auditors, billing, security etc. These all follow PDS guidelines and Microsoft best-practice guidance.

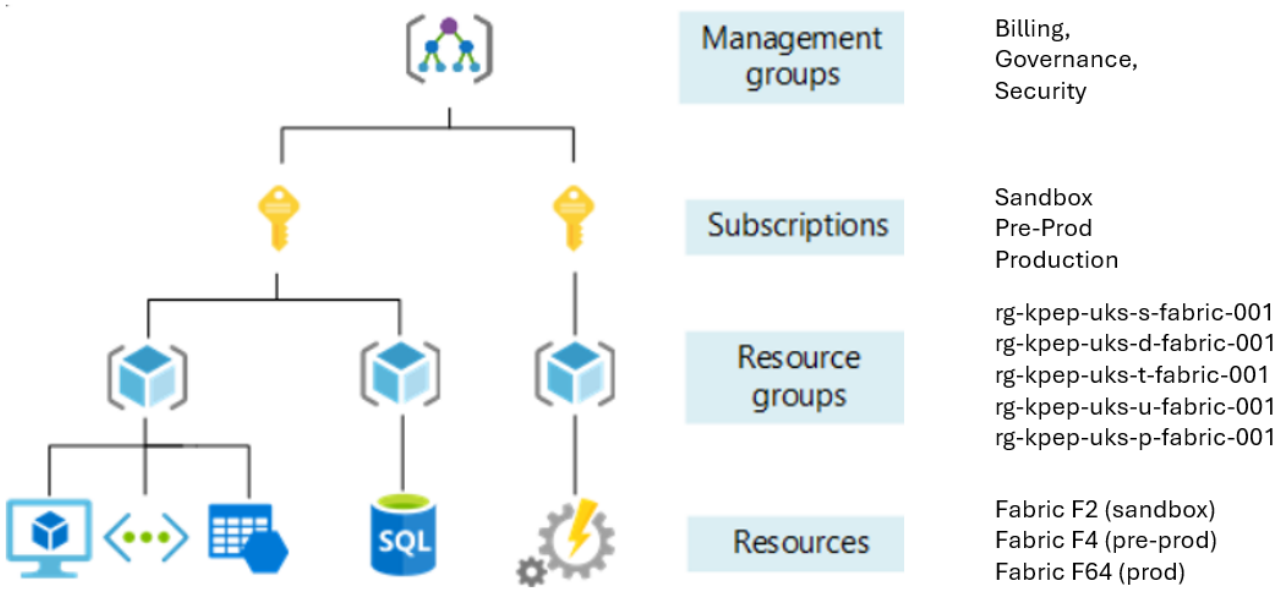


Figure 10 - Azure Access Control Hierarchy

2.1.7. Data Transformation Process, Flow and Connections

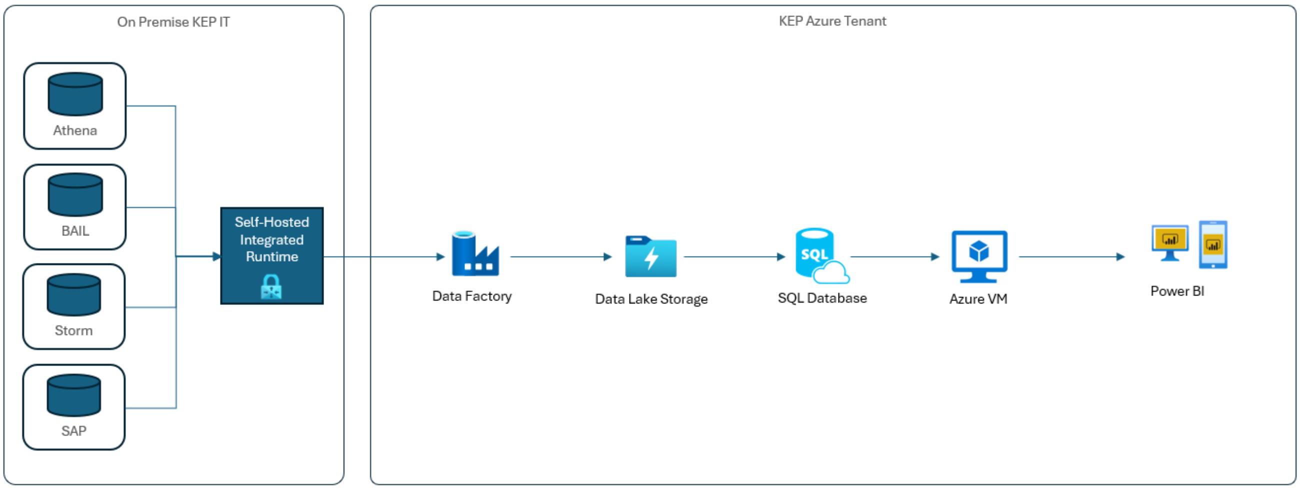


Figure 11 – Existing A4E to Source Connections

As shown in Figure 11, existing connections to source systems in A4E use a Self-Hosted Integration Runtime (SHIR). The existing A4E SHIR server will be decommissioned once all its data sources and pipelines have been migrated to the new environment. In the new architecture, we will require the following:

Component	Hosting location	Rationale	Comment
Azure Integration Runtime (AIR)	KEP Azure tenant	Where connectivity to cloud data is required – AIRs are secured connections from within Azure that don't require any additional security	At present there are no use cases - but this could be for connecting to external data sources such as external force reporting systems (TOEX), Companies House etc

		infrastructure from which to run.	
Data Gateway	KEP Data Centres	Where connectivity to on-premises data is required to Fabric.	This is the primary connectivity for connections to on-premises data - but as Fabric is relatively new, not all connections can be made from Data Gateways to on-premises data.
SHIR	KEP Data Centres	Where connectivity to on-premises data is required to Azure Data Factory (ADF).	This is in effect the legacy method to connect on-premises data to Azure and needs to be used where connectivity to Fabric is not yet possible through a Data Gateway - e.g. the connection to SAP. As new functionality is released in Fabric, data connections will be migrated to use the Data Gateway where possible.
SHIR Purview	KEP Data Centres	As Purview will scan both on-premises and Azure data sources, an on-premises SHIR server is required.	Microsoft stipulate that a Purview Integration Runtime cannot be shared with an Azure Data Factory Integration Runtime - therefore this requires its own SHIR server.
DevOps VM	KEP Azure tenant	A DevOps VM is required for deploying code in Azure and to run the data pipelines.	While there is an Azure managed option (Azure Hosted Agent Pipeline), this is a shared resource. For security, it is better to self-host a VM - this could be on-premises or in the cloud, but as there are no connections required to on-premises systems or data, it makes sense to host the DevOps VM in the cloud.

Figure 12 below shows the data connections between on-prem data sources and the development and live Azure environments, and the VMs required, as specified in the above table.

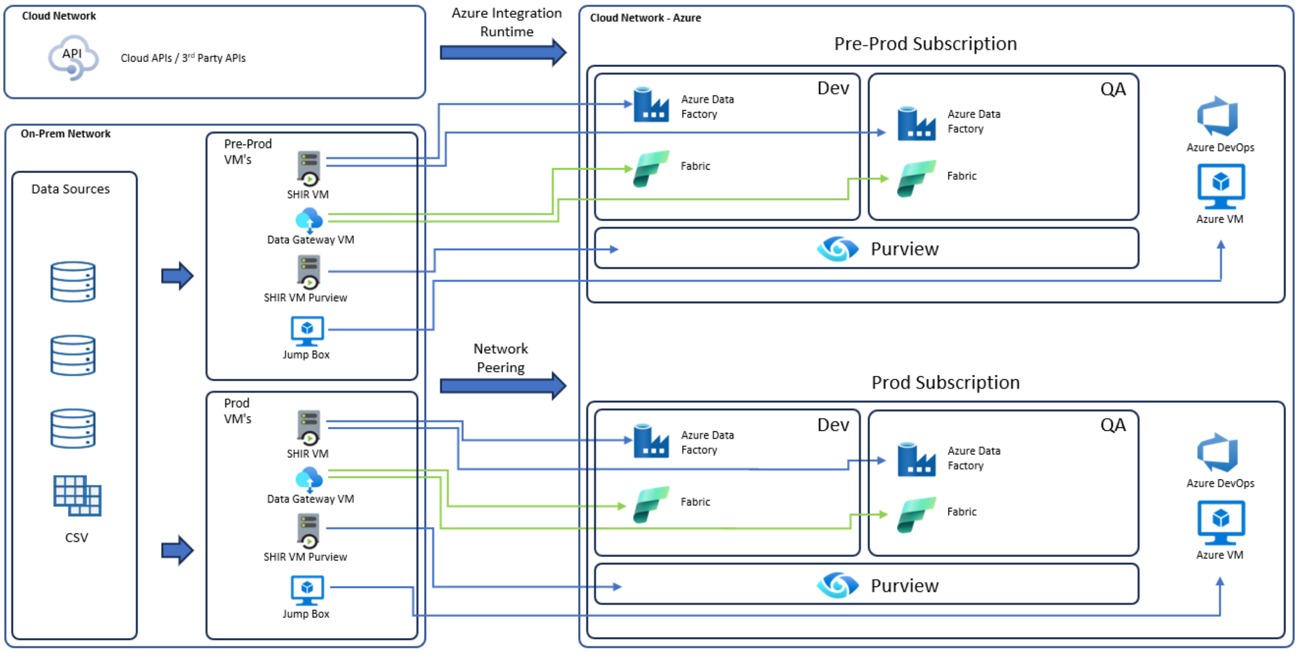
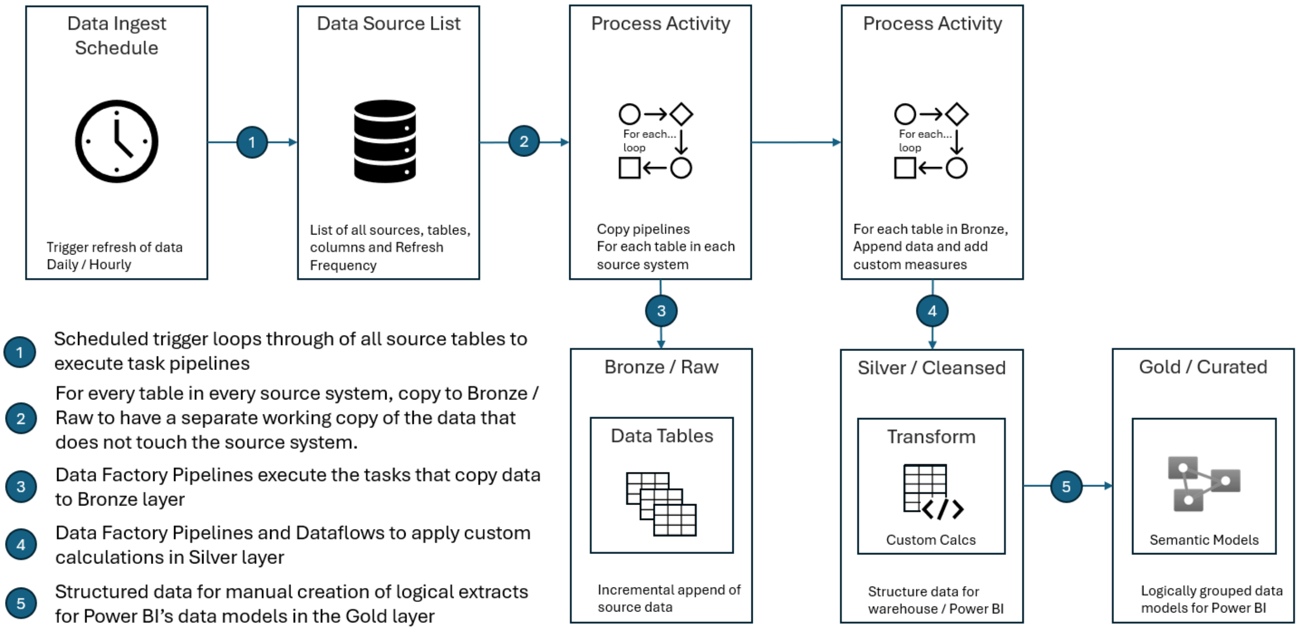


Figure 12 – Proposed Runtime Integrations

2.1.8. Metadata Framework


Data Ingest Schedule
Trigger refresh of data Daily / Hourly
1
Data Source List
List of all sources, tables, columns and Refresh Frequency
2
Process Activity
For each... loop
Copy pipelines For each table in each source system
Process Activity
For each... loop
For each table in Bronze, Append data and add custom measures
3
4
Bronze / Raw
Data Tables
Incremental append of source data
Silver / Cleansed
Transform
Custom Calcs
Structure data for warehouse / Power BI
5
Gold / Curated
Semantic Models
Logically grouped data models for Power BI

1 Scheduled trigger loops through of all source tables to execute task pipelines

2 For every table in every source system, copy to Bronze / Raw to have a separate working copy of the data that does not touch the source system.

3 Data Factory Pipelines execute the tasks that copy data to Bronze layer

4 Data Factory Pipelines and Dataflows to apply custom calculations in Silver layer

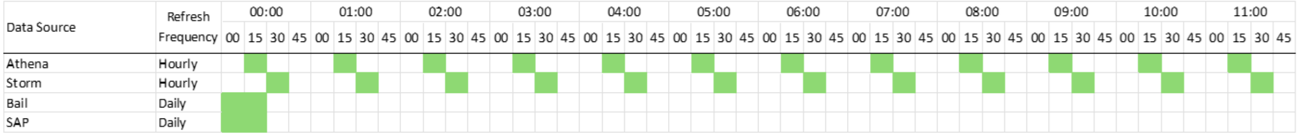
5 Structured data for manual creation of logical extracts for Power BI's data models in the Gold layer

Figure 13 - Metadata Framework Process Flow

The Metadata Framework (MDF) shown in Figure 13 above executes a sequence of commands that run data factory pipeline and custom calculation activities on the data, in Fabric, as it passes through the medallion layers. These activities loop through each field in each table from each source system's data (Design Decisions - DD04). This will structure the raw data that has been landed in the bronze layer into a common shared structured format, in the silver layer's process, that is suited to allow data to be related to each other in the gold layer's semantic models.

Each pipeline in the MDF is created for a specific data transformation task and reused by passing in the different source parameters for each copy task or custom transformation activity. This re-use of code removes the duplication of creating many multiple pipelines that would take a considerable amount of time to create manually as well as become unwieldy to manage and find for later updates.

Triggering of the MDF will commence during a period when the operational source systems are least used. For daily data runs this will typically be overnight. A schedule for each source system with its execution start time will be pulled into the MDF to action Fabric's data factory activities, an example as shown in Figure 14.

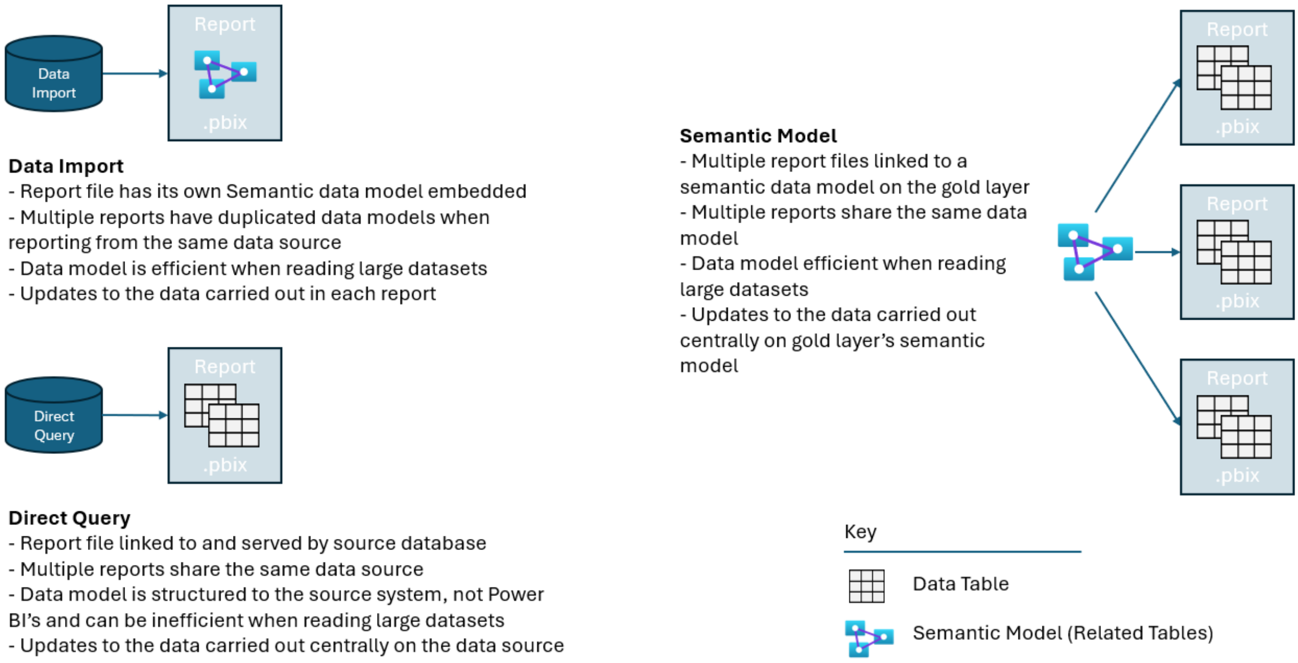


Data Source	Refresh	00:00				01:00				02:00				03:00				04:00				05:00				06:00				07:00				08:00				09:00				10:00				11:00			
	Frequency	00	15	30	45	00	15	30	45	00	15	30	45	00	15	30	45	00	15	30	45	00	15	30	45	00	15	30	45	00	15	30	45	00	15	30	45	00	15	30	45	00	15	30	45	00	15	30	45
Athena	Hourly		🟩				🟩				🟩				🟩				🟩				🟩				🟩				🟩				🟩				🟩				🟩				🟩		
Storm	Hourly			🟩				🟩				🟩				🟩				🟩				🟩				🟩				🟩				🟩				🟩				🟩				🟩	
Bail	Daily	🟩	🟩																																														
SAP	Daily	🟩	🟩																																														

Figure 14 – Example Data Refresh Schedule

Where more frequent refreshes of source data are required the MDF will be set to execute as often as is required to capture the updates to source data. These could be as frequent as every 15 minutes; a standard adopted when synchronising Microsoft's CRM product data. Such a frequency would work effectively on smaller data ingests that utilise delta checks over full truncate and reload of source tables.

2.1.9. Semantic Data Model



Data Import
- Report file has its own Semantic data model embedded
- Multiple reports have duplicated data models when reporting from the same data source
- Data model is efficient when reading large datasets
- Updates to the data carried out in each report

Direct Query
- Report file linked to and served by source database
- Multiple reports share the same data source
- Data model is structured to the source system, not Power BI's and can be inefficient when reading large datasets
- Updates to the data carried out centrally on the data source

Semantic Model
- Multiple report files linked to a semantic data model on the gold layer
- Multiple reports share the same data model
- Data model efficient when reading large datasets
- Updates to the data carried out centrally on gold layer's semantic model

Figure 15 - Semantic Data Model

Figure 15 above shows the typical methods for querying source data in Power BI. The Semantic model proposed in this design retains the best of both Data Import and Direct Query options, primarily by sharing updates to the underlying reports from the one model. Fabric by default connects directly from its OneLake to the semantic model - known as Direct Lake - for real-time refresh of the data in the models.

In consolidating the c.53,000 reports (see Design Decisions - DD01) from Business Objects the common features of reports that differ by date ranges / periods for example, will be served from the one data model that retains Role Based Access Control (RBAC) to present only the data required by each user from the same report, rather than having one report for each permutation.

The structure of the semantic models follows that of a data warehouse format, known as a Star Schema, that is better suited to reporting very large numbers of records. This is due to fewer hierarchical connections (relationship links) between tables than a structure that follows a traditional database model, which has many hierarchical connections. These star schema semantic models are separated from operational systems to avoid querying source data each time reports are run that would cause slowing in the operational systems and possible record locking that creates a high risk of data crashes and the almost inevitable resultant system crash. Each report would have to run against the source's multiple connections to query the data to present in the reports, creating significant read / write operations that would incur considerable but unnecessary cost every time the queries are run when opening a report.

The star schemas in the semantic models will have fewer relational connections and each semantic model's data feed has its data refreshed as part of the daily data ingest cycle. The models will have their assembled data cached in readiness to be displayed in Power BI thereby speeding up opening any of the attached reports to display the latest data.

A direct query report would refresh the query in the source database for each report that is attached to it. The read operations on the source grow proportionally to the number of attached reports and user-activated refreshes, which in turn would increase the cost to process of these refreshes. The Semantic Model does this once per scheduled refresh, as part of the refresh, pre-readying the data for users.

2.1.10. Governance - Purview

Microsoft Purview is a family of data governance, risk, and compliance solutions in Azure that help govern, protect, and manage the entire data estate. Purview gives an overview of where the data resides in source systems, wherever it is along the way when data is moving and being curated across the transformation process, and where it has ended up in the presentation environment e.g. Power BI.

The components of Purview that integrate natively with Fabric are:

- Purview Data Catalog - Automatically view metadata about Fabric items in the Data Catalog with a live view. This gives an overall method for users to search for data by column name, for example when needing to add to an existing report. The Catalog shows where repeat instances of the column being searched for originate from - in which source system - as well as where along the medallion flow the data has been added and transformed.

- Purview Information Protection - Allows discovery, classification, and protection of Fabric data using sensitivity labels from Microsoft Purview Information Protection. Sensitivity labels can be set on all Fabric items. Data remains protected when shared internally or externally, as permissioned via Role Based Access Control and AD / Entra. Compliance admins can monitor activities on sensitivity labelled data using Purview's Audit feature.

- Purview Data Loss Prevention (DLP) - DLP policies are currently supported in Fabric for Power BI semantic models only. DLP policies detect upload of sensitive data into semantic models. They can detect sensitivity labels and sensitive info types, such as credit card and social security numbers. They can be configured to generate policy tips for semantic model owners and alerts for security admins. DLP policies can also be configured to allow data owners to override them for specific use cases that might otherwise be too restrictive.

• Purview Audit - Fabric user activities are logged and made available in Purview's audit log.

Purview sits underneath the entire source through transformation to destination process, underneath the entirety of Fabric, as a security safety net.

2.2. Business and Data Architecture

Business and Data Architecture	
Information Asset Owner	Dr Natalie Mann and Nicola Endacott, as Heads of Data Analysis teams respectively for Essex and Kent, will be the IAOs for the Data Transformation platform, once permission to process the data has been granted by the source system IAOs. Individual SyOps and DPIAs will need to be reviewed / amended for each data source as each will have their own SyOps, DPIAs and requirements. Example: One SyOP for Athena which would be signed off by Athena IAO + Data transformation IAO. Athena SyOP then could be updated to include any new Athena data use cases within data transformation.
Which Force	Kent Police and Essex Police.
Who uses it	There are currently: - 467 Report Writers capable of creating and amending reports - 2,530 Guest users capable of interacting with reports such as filtering and exporting data - 2,509 Viewers that can only view report content Daily between 7am and 5pm there are between 50 and 100 individual consumers of BI Launchpad at any one time. The daily average of total unique consumers in the last year per month is 250. The approximate numbers per team are shown below: - Athena Management Organisation / Athena Development Team (6) - Business Services (13) - Corporate Finance (19) - Central Analytical Team (67) - FCIR (6) - Human Resources/Learning and Development (10) - Health Services (5) - Performance Analysis Unit/Intel (49) - Professional Standards Department (5) - Data Analysis teams - Data Engineering team - Existing Business Objects users - Data / report consumers (includes external third parties*) *N.B. The full / exact user requirements for the Data Transformation project, as a whole, are still being defined. It is expected this will include the provision of reports and / or data to external third parties. Once the requirements are fully understood, the system will be designed, documented and tested accordingly - with the HLD updated for approval.

What do they use it for	Used to enhance operational and organisational decision making and reporting, with the aim of improving efficiencies and reducing crime.
Data Classification	GSC Official Sensitive. Additional data classification tags for customising governance controls can be auto assigned by Purview as well as custom classifications assigned by data owners. E.g. auto tagging of Post Codes, Phone Numbers, and custom tagging Person Code (e.g. from Athena), Crime Reference Number etc. Secret and above data will not be held or processed in this new solution.
Data Retention and Weeding	Data retention will need to be reviewed on a case-by-case basis as each source system is ingested into the platform. The bronze layer will hold all ingested data, with the most recent 30 days' worth of data loaded in daily, per scheduled refresh. There will be full truncate and reload carried out monthly. The silver layer is the cleansing area where any given period's ingested data types are standardised. The data is transient for the curation process of data from bronze to gold. Therefore, no backup is required. The gold layer is the warehoused copy of all historic data grouped into logical lakehouses that the reports require. See Figure 3 - Medallion ETL Data Load Pattern. It is the data held in bronze and gold layers that therefore needs to be considered for data retention and weeding. Recommendation for weeding is to follow the protocols applied against each source system. These would naturally pull through to bronze in the initial part of the data ingest process, to follow the data weeding that happens at source. In the gold warehouses an historic record is retained, using the principle of archiving the changed record and flagging the latest change to it as current. Any weeding that is to be applied to a record will need to be applied to each record's history too. This will be carried out by overwriting any personally identifiable parts of a record.
Exit Strategy	The data in Fabric will be structured according to industry standard design principles - the medallion architecture. The ingested data will be retained in parquet file format. These are system agnostic therefore moving to a different reporting system, ETL / ELT process too, could be as simple as a plug and play of this data.
Recovery Point Objective (RPO) **The time between backups**	Azure Fabric is a high-availability platform, but Microsoft do recommend customers perform their own backups. As the platform is built this will be investigated further, agreed with all stakeholders, tweaked as necessary, and the HLD updated. Broadly speaking there will be five tiers of data held in the system: 1) Report config, source code repository etc 2) Raw data extracted from source systems (typically bronze data) which can easily be rebuilt from the source systems 3) Raw data only held in the platform - e.g. where data has been BRC'd into the platform and the source system deleted 4) Curated data per refresh (silver layer)

	5) Warehoused data - assembled, structured and grouped for Power BI reporting (gold layer) In Business Objects, the current RPO is 24 hours - it is anticipated this level will be maintained and likely improved in the new platform. With the latency of replication of data from primary to secondary zone, Microsoft quote typical RPO for this of 15 minutes. Further detail provided in 2.2.1 below.
Recovery Time Objective (RTO) Target time for a recovery after a disaster.	As per above, the RTO will be investigated further, agreed with all stakeholders, and the HLD updated. In Business Objects, the current RTO is 2-3 days - it is anticipated this will be improved in the new platform as a result of the extensive and consolidated nature of backups and their proximity to the platform. Zonal redundancy has been selected as a means to keep recoverable copies of the data platform, that are themselves built with redundancy, whilst maintaining the geo-location requirements specified by the PDS. Further detail provided in 2.2.1 below.
Data Migration / Back Record Conversion (BRC)?	Part of the work to identify the data sources that need ingesting into the new Data Transformation platform will include identifying any data that is held within SAP Business Objects that needs to be migrated into the new platform. Examples may include data that has been BRC'd into Business Objects from a legacy system, to allow the legacy system to be decommissioned.
Legacy System replacement?	SAP Business Objects SAP Data Services Analytics 4 Everyone (A4E) - the Data Transformation project will lead to the removal of the existing underlying architecture for A4E, but the A4E dashboards will remain available to the end users, with their data served from Microsoft Fabric.

2.2.1. Backup and Data Resilience

Microsoft recommends using Zone Redundant Storage (ZRS) in the primary region for Azure Data Lake Storage workloads, Fabric's OneLake in the design here. This copies data synchronously across three Azure availability zones in the primary region.

The primary region for the data platform and everything in Azure associated to it is UK South, as stipulated in the PDS design.

For applications requiring high availability (HA), Microsoft recommends using ZRS in the primary region, and additionally replicating to a secondary region. The secondary region would be UK West. In order to maintain high availability and to support RTO in the most efficient possible manner, our design will incorporate replication to the secondary region. See Figure 16 below.

Fabric's OneLake uses zone-redundant storage (ZRS) in UK South. ZRS provides fault tolerance to datacenter failures by copying data synchronously across three Azure availability zones in the primary region. ZRS offers a durability of at least 99.9999999999% (12 nines) over a year. This equates to fractions of a minute in possible downtime.

Because data is replicated to the secondary region asynchronously, a failure that affects the primary region (and all 3 availability zones) may result in data loss if the primary region cannot be recovered. The interval between the most recent writes to the primary region and the last write to the secondary region is known as the recovery point objective (RPO). The RPO indicates the point in time to which data can be recovered. The Azure Storage platform typically has an RPO of less than 15 minutes, although there's currently no SLA on how long it takes to replicate data to the secondary region. Within each region, availability zones are connected through a high-performance network. Microsoft strives to achieve an inter-zone communication with round-trip latency of less than approximately 2 milliseconds (ms). Round trip latency is affected by distance between zones. UK South to UK West's metrics are shown to be 7ms.

Fabric is a full SaaS product that manages all aspects of its components itself. Only the data is the responsibility of the user. Business Continuity and Disaster Recovery (BCDR) is Fabric's implementation of ZRS and is enabled in the Fabric admin portal.

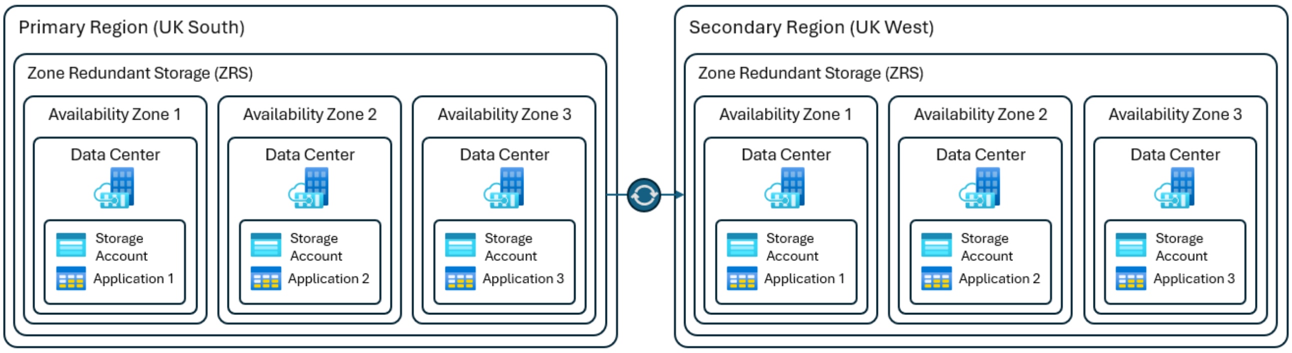


Figure 16 - Zone Redundant Storage with High Availability

2.2.2. Backup Schedule / Periods

Resource	Backup Frequency	Location	Retention Period	Notes
Fabric	Instant / Realtime	Redundancy Zone	90 Days	Built-in feature
Meta Data Framework	Instant / Realtime	Azure Git	Indefinite	Git Repository and pipelines in Fabric
Bronze	Daily	Fabric	Indefinite	After each daily load Subject to retention periods of personal data and weeding to overwrite personally identifiable data
Silver	N/A	Fabric	N/A	Transient data for daily processing only. No backup needed
Gold	Daily	Fabric	Indefinite	After each daily load Subject to retention periods of personal data and weeding to overwrite personally identifiable data
SHIR VMs	Daily	Commvault / Isilon	30 Days	For ADFs and Purview in Pre-Prod and Prod
Data Gateway VMs	Daily	Commvault / Isilon	30 Days	For Fabric in Pre-Prod and Prod

DevOps VM	Daily	Azure Backup	30 Days	DevOps DevOps agent only to reside on the VM
DevOps Git	Instant / Realtime	Azure Storage	Indefinite	No separate backup needed. All handled by Microsoft / DevOps

2.2.3. Data Metrics

Department	Number of reports run / month	Number of records returned	Total size of files
All	55,353 (Dec '24)	5.3 billion	74.22 GB

Note. These figures only cover BI Launchpad data consumption and not the data that is ETL'd via SAP Data Services

2.2.4. Management Information Reporting requirements

Microsoft Fabric and the underlying Azure infrastructure includes comprehensive reporting capabilities. As the platform is built and starts to get used, the exact reporting requirements will be reviewed, but are expected to include requirements such as:

- Reporting for billing, which is built into all Azure products. This will be available in each subscription enabled for the data transformation project. All cost analysis log data will be further categorised by resource group.
- Operational performance measurement and monitoring Azure's built in metrics and logs, which typically are retained for 90 days, will be utilised by the Metadata Framework (Design Decisions - DD04) to identify and record the identifiers of pipelines that have been run so that they can be used to trace errors and re-executed in the event of pipeline execution failure.
- Log Analytics, which is an Azure resource that further enhances the availability of Azure usage metrics and logs that can be stored within the Log Analytics part of Azure, in the Management Subscription of our Enterprise Landing Zone (Figure 4 - Page 14), presented through Power BI dashboards alongside the warehoused data on the gold layer.
- Azure Network Watcher is enabled when creating or updating a virtual network (must be actively switched off).

2.3. Application Architecture

2.3.1. Interfaces to Line of Business Applications and Core Infrastructure

This section will be updated as the project progresses. The project will rely on a number of elements of our Infrastructure - e.g. AD will be used for authentication and access, Outlook and Teams can be used for metadata framework messaging (e.g. failed pipelines), where reports may still get emailed out etc (although ideally emails will be the exception as we want people to subscribe to reports and take responsibility for viewing the data they require).

Name	Source	Target	Method	Frequency	Comment
Data Gateways	Kent & Essex on-prem systems	Fabric	New VMs for Pre-Prod and Prod environments (see DD05 and Figure 9)	To be determined by refresh frequency of source system data	Looking to automate the spin-up on demand and corresponding spin-down in accordance with refresh frequencies

2.3.2. Identity and Access Management (AD, SSO etc.)

This section will be updated as the project progresses. Using a permissions and access hierarchy matrix, the project will identify the different types of user (e.g. break-glass, billing, data pipeline engineers, data scientists, analysts, end users, PSD, external users etc) and how their granular levels of access will be governed and audited - this is expected to primarily use the inbuilt capabilities such as Azure Active Directory/Entra ID, as well as tools like Purview that this project will introduce.

Identity and Access Management	
How is access managed?	Azure Active Directory (AD) / Entra Role Based Access Control (RBAC) for the data Purview data tagging for governance and access management Azure VM: Data Engineers will need to push deployment code scripts
New or existing AD groups?	Leveraging existing AD groups for retention of access control where possible. For groups associated to reports and data output from Business Objects these will be reconfigured to new groups that will be designed to replicate existing logic.
Describe the JML process	Joiners, Movers, Leavers (JML) handling will follow organisation hierarchy in SAP / AD alongside existing protocols. These will be associated to the AD / Entra groups and associations for each JML. Personas will assist in bridging functionality that resources require to access and operate on the platform with the access permissions and PIM roles available. Example: Data Engineer persona will require all contributor role assigned to grant permissions in the Fabric environment. This will need granting / revoking to anyone joining / leaving the Data Engineering team
Auditing capability	The data audit features of Purview provide a forensic level detail of access and sharing of reports. Access to this data is controlled by RBAC and governed by Purview. The audit data is held for 90 days by default but can be retained for longer, with Audit Premium. Data Engineering auditing will be baked into the metadata framework with associated metrics and management in dedicated Power BI reports.

2.3.3. Licensing

Licensing Model	
Relies on existing licenses?	Kent and Essex currently have 110 Power BI Pro licenses. The majority of these will no longer be required, but a small number may need to be retained where required - for example for the consumption of external partner's dashboards (e.g. TOEX). The existing subscription to Power BI Premium (A4E Essex) will be replaced by Fabric F64 for both Kent and Essex, transitioning upon expiry of the existing A4E support contract in November 2025 to the end of A4E's Power BI Premium subscription commitment in March 2026. Data model sharing and B2B licensing is currently being investigated, to eliminate the need for additional (paid) Power BI Pro licenses The current E3 Microsoft license includes a license for every user to consume Power BI reports. This is by way of a Power BI Free license, effectively as a viewer / read only means to gain access to Power BI content. VMs for SHIRs and Data Gateways will use KEP licenses for Windows under enterprise licensing. The Azure Dev Ops VM will use a KEP Hybrid Windows enterprise agreement license, effectively a Bring Your Own Licence (BYOL) and 3-year reservations
Requires new licenses?	Fabric F2 Tier - Sandbox (PAYG on demand) Fabric F4 Tier - Pre-Prod Environment Fabric F64 Tier - Prod Environment Purview plus Audit Premium. Purview is a PAYG licence. Audit Premium is per-user based upon user activity being scanned Azure Monitor - Enhanced logging and usage / monitoring Network Watcher - Enabled by default when virtual networks created Log Analytics - collection of metrics for operational reporting and analysis Storage Accounts - Archive and retention (2.2 - Business and Data Architecture - Data Retention) Production / Live usage will be monitored to ensure appropriate tier is selected. As usage increases the tier can be increased accordingly. Downgrading can only be done at the end of a subscription period
Application specific licenses?	Visual Studio Code (free MS download) installed on jump box for connection to developer laptops, which will be used for building and managing deployment Pipelines and IaC in DevOps, from client devices. VS Code uses plugins for Python, R and DevOps deployment etc. to modularise its use rather than the fully featured Visual Studio. Azure Data Studio (free MS download) will be used for data querying and general writing and reading of SQL scripts. Supplement for SQL Server Management Studio (SSMS) for more cloud-centric database work OneLake File Explorer (free MS Download) will be primarily used by the data scientists when uploading files to fabric OneLake that wouldn't require passing through the ETL process Each Azure subscription comes with a 5 user DevOps license which will be utilised for project management and sprint planning, and activation of the DevOps deployment operations.

2.4. Technology Architecture

Server & Storage Architecture as described at 2.1.7 and shown in Figure 12 – Proposed Runtime Integrations the current A4E SHIR server will be replaced by the following servers.

On-premises VMs:

Server/ Location	Purpose	Quantity	Physical Virtual	CPUs	RAM	Storage	Software/OS
epvmdtshir01apd / Spring Park	New SHIR server for on-premises data connectivity to Pre-Prod ADF	1	V	4	8	C: 80 GB	Windows 2022 .NET Framework 4.8.2 or later
epvmdtshir01app / Spring Park	New SHIR server for on-premises data connectivity to Prod ADF	1	V	4	8	C: 80 GB	Windows 2022 .NET Framework 4.8.2 or later
epvmdtshpu01apd / Spring Park	New SHIR server for connectivity to Pre-Prod Purview	1	V	4	8	C: 80 GB	Windows 2022 .NET Framework 4.8.2 or later
epvmdtshpu01app / Spring Park	New SHIR server for connectivity to Prod Purview	1	V	4	8	C: 80 GB	Windows 2022 .NET Framework 4.8.2 or later
epvmdtfadg01apd / Spring Park	New Data Gateway for Pre-Prod Fabric connectivity to on-prem data sources	1	V	4	8	C: 80 GB	Windows 2022 (or 2025 if Building Block is ready in time) .NET Framework 4.8
epvmdtfadg01app / Spring Park	New Data Gateway for Prod Fabric connectivity to on-prem data sources	1	V	4	16	C: 80 GB	Windows 2022 (or 2025 if Building Block is ready in time) .NET Framework 4.8
Totals		6	6 VMs	24	56	480 GB	

Service level	Gold*

Azure VMs:

Server/ Location	Purpose	Quantity	Physical Virtual	CPUs	RAM	Storage	Software/OS
vmkpepddtdvops/ Azure UK South	New Azure DevOps Agent for PreProd Azure DevOps CI/CD pipelines	1	V	4	16	128 GB	Azure Spec: Standard_D4as_v6 Windows OS (BYOL available)
vmkpeppdtdvops/	New Azure DevOps Agent for Prod Azure	1	V	4	16	128 GB	Azure Spec: Standard_D4as_v6

Azure UK South	DevOps CI/CD pipelines						Windows OS (BYOL available)
Totals		2	2 VMs	8	32	256 GB	

DevOps Jump Box:

Server/ Location	Purpose	Quantity	Physical Virtual	CPUs	RAM	Storage	Software/OS
Azure UK South	New servers for Data Transformation Pre-Prod and Prod. Jump box with software not available on laptops	2	V	4	16	C - 80GB D - 50GB	Windows Server 2022

Storage Type	Purpose	Volume	Growth rate	Replicated?	Backed Up?
N/A	N/A	N/A	N/A	N/A	N/A

2.4.1. Network Architecture

This section will continue to be updated as the project progresses. The Network team will be closely involved as the networking requirements become clearer.

Source	Target	Port	Comment
Inbound SHIR connections			
epvmdtshir01apd **epvmdtshir01app**	kphqgen04sqp.netr.ecis.police.uk	1433	Risk Register
	kphqgen04sqp.netr.ecis.police.uk	1433	Bail
	kphqoda2-scan-lsnr.netr.ecis.police.uk	1521	STORM Kent
	ephqoda2-scan-lsnr.netr.ecis.police.uk	1521	STORM Essex
	prdsapci.ecis.police.uk	443, 3200, 3300, 3600	SAP
	ephqoda2-scan-lsnr.netr.ecis.police.uk	1521	Athena
Inbound Data Gateway connections			
epvmdtfadg01apd **epvmdtfadg01app**	kphqgen04sqp.netr.ecis.police.uk	1433	Risk Register
	kphqgen04sqp.netr.ecis.police.uk	1433	Bail
	kphqoda2-scan-lsnr.netr.ecis.police.uk	1521	STORM Kent
	ephqoda2-scan-lsnr.netr.ecis.police.uk	1521	STORM Essex

	prdsapci.ecis.police.uk	443, 3200, 3300, 3600	SAP
	ephqoda2-scan-lsnr.netr.ecis.police.uk	1521	Athena
Inbound SHIR for Purview connections			
epvmdtshpu01apd **epvmdtshpu01app**	kphqgen04sqp.netr.ecis.police.uk	1433	Risk Register
	kphqgen04sqp.netr.ecis.police.uk	1433	Bail
	kphqoda2-scan-lsnr.netr.ecis.police.uk	1521	STORM Kent
	ephqoda2-scan-lsnr.netr.ecis.police.uk	1521	STORM Essex
	prdsapci.ecis.police.uk	443, 3200, 3300, 3600	SAP
	ephqoda2-scan-lsnr.netr.ecis.police.uk	1521	Athena
External FQDNs used by SHIR and Data Gateway			
epvmdtshir01apd **epvmdtshir01app**	adf-kpep-uks-dtdev.datafactory.azure.net	443	The SHIR server will be registered to this Data Factory. Required by the SHIR to connect to the ADF service.
epvmdtshir01apd **epvmdtshir01app**	*.servicebus.windows.net	443	Required by the self-hosted integration runtime for interactive authoring. Interactive authoring is required to test the connection, browse folder / table list, get schema and data preview.
epvmdtshir01apd **epvmdtshir01app**	download.microsoft.com	443	Required for SHIR auto-updates. It is recommended this is implemented.
epvmdtshir01apd **epvmdtshir01app**	*.azuredatalakestore.net	443	Required only when copying from or to Azure Data Lake Store. When we connect to the data via Data Factory
Outbound Ports			
Data Gateway			
epvmdtfadg01apd **epvmdtfadg01app**	*.servicebus.windows.net	443 and 9350-9354	Required by the data gateway for interactive authoring. Listens on Azure Relay over TCP. Port 443 is required to get Azure Access Control tokens.

epvmdtfadg01apd **epvmdtfadg01app**	download.microsoft.com	443	Required by the data gateway for downloading updates. If auto update is disabled, this config can be skipped
epvmdtfadg01apd **epvmdtfadg01app**	*.core.windows.net	443	Used by the data gateway to connect to the Azure storage account when using the 'staged copy' feature
epvmdtfadg01apd **epvmdtfadg01app**	Inbound ports PowerShell	8060 (TCP)	Required by PowerShell encryption cmdlet and by the credential manager application to securely set credentials for on-premises data stores on the data gateway.
epvmdtfadg01apd **epvmdtfadg01app**	*.powerbi.com	443	Used to identify the relevant Power BI cluster
epvmdtfadg01apd **epvmdtfadg01app**	*.analysis.windows.net	443	Used to identify the relevant Power BI cluster
epvmdtfadg01apd **epvmdtfadg01app**	*.login.windows.net, login.live.com, aadcdn.msauth.net, login.microsoftonline.com, *.microsoftonline-p.com	443	Used to authenticate the gateway app for Microsoft Entra ID and OAuth2. Note that additional URLs could be required as part of the Microsoft Entra ID sign in process that can be unique to a tenant
epvmdtfadg01apd **epvmdtfadg01app**	*.msftncsi.com	80	Used to test internet connectivity if the Power BI service can't reach the gateway
epvmdtfadg01apd **epvmdtfadg01app**	*.dfs.fabric.microsoft.com	443	Endpoint used by Dataflow Gen1 and Gen2 to connect to OneLake
epvmdtfadg01apd **epvmdtfadg01app**	*.datawarehouse.pbidedicated.windows.net	1433	Old endpoint used by Dataflow Gen2 to connect to the Fabric staging lakehouse
epvmdtfadg01apd **epvmdtfadg01app**	*.datawarehouse.fabric.microsoft.com	1433	New endpoint used by Dataflow Gen2 to connect to the Fabric staging lakehouse
epvmdtfadg01apd **epvmdtfadg01app**	*.frontend.clouddatahub.net	443	Required for Fabric Pipeline execution
epvmdtfadg01apd **epvmdtfadg01app**	*.fabric.microsoft.com	TCP 443	Fabric Portal
SHIR			
Should all this below be separated as it's the Fabric connection info ?			
OneLake			
DFS APIs	*.onelake.dfs.fabric.microsoft.com	TCP 443	Default OneLake endpoint Data File Storage on lakes

Blob APIs	*.onelake.blob.fabric.microsoft.com	TCP 443	Binary Large OBject - all the standard data files and images
Workstations and Laptops Allow List - Devices to connect to the Fabric IDE			
No specific endpoints other than the customer's data store endpoints required in pipelines and behinds the firewall			User can use service tag DataFactory, regional tag is supported, like DataFactory.WestUS
Lakehouse **https://cdn.jsdelivr.net/npm/mona co-editor***		443	
EUC Devices	http://res.cdn.office.net/	443	Notebook Inbound Connection - Icons
EUC Devices	https://*.pbidedicated.windows.net wss://*.pbidedicated.windows.net (HTTP/WebSocket)	443	Notebook backend
EUC Devices	https://onelake.dfs.fabric.microsoft.com	443	Lakehouse backend
EUC Devices	https://*.analysis.windows.net	443	Shared backend
EUC Devices	https://pbides.powerbi.com	443	DE / DS extension - UX
EUC Devices	https://aznb-ame-prod.azureedge.net https://*.notebooks.azuresandbox.ms https://content.powerapps.com https://aznbcdn.notebooks.azure.net	443	Notebooks - UX
EUC Devices	http://res.cdn.office.net/	443	Spark Inbound connections - Icons
EUC Devices	https://pypi.org/*	443	Library management for PyPI
EUC Devices	https://pypi.org/*	443	Data Science Library management for PyPi
EUC Devices	https://*.z[0-9].kusto.fabric.microsoft.com	443	KQL Database - Store for real-time data and event streams
EUC Devices	sb://*.servicebus.windows.net	http: 443 amqp: 5672/5673 kafka: 9093	Eventstream - Feed for realtime data

2.4.2. End User Device Architecture

Client architecture	
Browser based?	Microsoft Edge will be the primary method of access for end users to consume reports. While reports can be created and modified using Power BI in the browser, the desktop tools listed below are more feature rich and more appropriate for engineering tasks.
Thick client?	Power BI Desktop – this is the development tool for reports and dashboards and will be used by the Analysts and creators and modifiers of Power BI reports. Power BI Report Builder - this is the development tool for paginated reports and will be used by the Analysts and creators and modifiers of paginated reports. Specialist reports that are currently served using PowerPoint will require recreation with direct connections to the data visuals in Power BI. There is an ongoing discussion with security, and analysis of the Power BI Plugin for PowerPoint to determine how best to make the plugin available.
Mobile requirement?	A native Android app exists for Power BI and will be required on force mobiles. While all reports will render on a mobile, development effort is required to optimise them for mobile use at the time of report creation, if flagged as being required.
Peripherals?	None

2.5. Security Architecture

This section will be updated as the project progresses.

Security architecture	
Data in Transit	Data in transit between Microsoft services is always encrypted with at least TLS 1.2. Fabric negotiates to TLS 1.3 whenever possible. Inbound Fabric communication also enforces TLS 1.2 and negotiates to TLS 1.3, whenever possible. Microsoft Documentation Data passing through the SHIRs and Data Gateway VMs will all be encrypted as above.
Data at Rest	All Azure data is encrypted at rest by using Microsoft-managed keys - this will be to a minimum of AES 256. Microsoft Documentation. No data resides in the VMs used for the SHIRs and Data Gateways, and in keeping with current KEP Technical Standards the disks will not be encrypted.
IP Allow Listing	Details of any allow listing used will be documented here once known/agreed.
Patching	VMs will be patched in line with existing policies and protocols. On Prem - Ivanti Azure - Updated using Azure's patching mechanisms
Malware protection	VMs on prem will be utilising existing BitDefender protection. Azure VMs will utilise Windows Defender
Penetration Test Required?	Yes.

	The exact details of the pen test will be agreed with Information Security and be performed, with all relevant vulnerabilities addressed, prior to holding live data in the environment.
Monitoring	Details of the monitoring used will be documented here once known/agreed.
Certificates Required	Details of the certificates required for the solution will be documented here when known.

Name	Purpose	Installation Location	Validity	Certificate Authority	New or Existing?

3. Design Decisions

Key design considerations are based upon increasing efficiency to make data available for reporting through standardisation and adoption of best practices. Costs, both in re-engineering the transformation and reporting processes are a major influence on the choices of development frameworks / methodologies chosen, very much with consideration to ongoing usability and support post-project.

ID	Decision	Rationale	Impact
DD01	Consolidate reports to logical groups	Currently there are circa 53,000 reports many of which are duplicates of the same report with minor changes that are saved-as copies of the original. Where possible, single reports will be created, served from a central data model that has its data access constrained by user role / permissions - users will then be able to tailor the information they view in the report without having to create bespoke reports for every user/scenario.	Significantly reducing the number of reports required will have multiple benefits with little to no user impact.
DD02	Kent and Essex being able to share more of the data	Whereas currently A4E only serves Essex data/users, the platform will be shared across both Kent and Essex.	A single combined modern data reporting and analytics platform used and managed by both forces, for both forces.
DD03	Update transformation architecture to medallion	Initial load of source system data to raw / Bronze results in one-time read of the data, that will also create an auditable and archivable copy of system data. Curation of the data will be carried out separately in silver and gold layers.	Reduction in read operations reduces load and possible contention on data sources. Rolling archival to low cost (cold) Data Lake storage. Separating ingest from curation gives granular level approach to read / write processes.
DD04	Metadata Framework for script-driven	Many source systems with multiple data tables require multiple data factory pipeline activities to pull data into the transformation	Significant reduction in the time it would take to create the pipelines, by looping and re-

	transformation pipelines in Fabric Data Factory	(medallion) layers. Scripting the creation of these pipelines rather than hard coding them in data factory allows re-use of code and significant reduction in development time.	using scripted data factory commands. Reduction in possibility of coding errors by running source code from the framework rather than recreating pipelines manually.
DD05	Self-Hosted Integration Runtime (SHIR) instead of Azure Integration Runtime (AIR)	AIR used for cloud-to-cloud connections in Azure Data Factory, SHIR used for connection from Azure to on-prem data sources. With the majority of data sources being on-prem SHIR will be used over AIR.	Self-Hosted Integration Runtimes are an established part of the KEP technical landscape, so the impact is minimal.
DD06	Anonymised Data for Development work	In lieu of pen testing, data extracts required for the migration of A4E will have personally identifiable data anonymised. These will be uploaded to the Pre-Prod Dev area for data engineering Data structure will be retained and row counts constrained to permit lower tier of Fabric in development environment.	Development work can continue before the pen test without providing unvalidated connections back to the sources. Post pen test the data will be refreshed periodically for ongoing development use
DD07	Self-Hosted Dev Ops Agent running in the Azure VM instead of Azure Hosted Dev Ops Agents	Self-hosted Dev Ops Agents allow us to be in control of how and when we update the Dev Ops Agent and gives us a better guarantee of performance as Dev Ops is hosted on a VM owned and controlled by us which only processes our tasks.	Requires in-house built Azure VM that is managed and supported by us. Future expansion to run concurrent jobs would also be less expensive on an in-house Azure VM than on a public Dev Ops service.
DD08	Adopting a centrally managed artefact-based repository of data engineering code and Power BI reports. Using script (YAML) based deployment tasks that will be actioned by following the CI/CD process	Utilising Azure DevOps code repositories and deployment pipelines to manage all development and updating tasks from DevOps code repo. Script-based deployment will allow reuse for all updates and will only deploy to the designated environments: dev -> QA -> UAT -> live	All BI reports will be deployed from centrally version-controlled sources that will overwrite any local adjustments made. Reduction / elimination of the c.53,000 reports that have accumulated over years of making tweaks to copies that are tweaks of copies
DD09	Monitoring and Documenting in Azure	Setting the Azure monitoring equivalent to on-prem to match current infra monitoring. Standardises Azure and on-prem monitoring and reporting	May require evaluation of logged information to avoid unnecessary cost uplift

4. Solution Building Blocks

BB Name	(N)ew (E)xisting	Functional Area	Reference	Rationale

5. Implementation Tasks

Task Description	Who / Which Teams
Design of permission groups and roles	IT / Business / EUC Dev
Create resource groups, Pre-Prod environments and install Fabric capacities	IT*
Create resource groups, Prod environments and install Fabric F64	IT*
Identify initial and subsequent data sources to be ingested into the new environment. Initial data sources expected to be Bail and SAP as the first two data sources used by A4E	Data Engineering Team / Business
Determine the data that will be permitted for development as well as what will require anonymisation	Data Engineering Team / InfoSec
Build and implement the initial connectivity (e.g. Bail and SAP) ready for initial testing	Data Engineering Team / IT* / Business (likely to require Networks for connectivity and Database for access controls)
Creation of Servers (VMs) for connectivity to Azure Data Factory, Fabric and Purview. Installation of Agents and Runtimes on VMs	Server & Networks Teams / Data Engineering Team
Install Purview in Pre-Prod	IT
Configure Purview Connect main Azure Purview to Pre-Prod Environment Set up scanning Apply Tagging rules	IT (Architecture) / EUC Dev Servers IT (Architecture) IT (Architecture)
Engineering of the Metadata Framework to capture source data and make available logging and monitoring information.	Data Engineering Team
Implement the management information reporting requirements – such as Log Analytics, Purview etc	Data Engineering Team
Pen Testing, support and remediation activities	InfoSec / IT*
Rollout of Power BI reports and dashboards, specifically new world access	Data Engineering Team / IT* / Analysts
Periodically review the firewall rules and connections to ensure all the requested firewall rules are actively being used (e.g. as functionality becomes available in Data Gateway and we move traffic from the SHIR to the Data Gateway - can we close firewall rules/ports?)	Network Team/Data Engineering Team

*the exact roles and responsibilities of the different IT Teams (e.g. Server and Infrastructure, EUC Dev, Networks, Database, Data Engineering Team etc) within the KEP Azure tenant will need to be agreed and appropriate training and time allowed for any upskilling etc.

6. Technical Risks / Issues, Assumptions / Dependencies

This section should address key TECHNICAL Risks (RR), Issues (II), Assumptions (AA) and Dependencies (DD). PMO should hold the master Risk log for the project.

Type	Description	Impact
RR01	For clarity the network traffic colour legend only indicates the data encryption in transit state for the predominant network traffic. **Strongly encrypted traffic GREEN – TLSv1.2, similar or greater. For IPsec meets or exceeds the NCSC "recommended" profile Using IPsec to protect data - NCSC.GOV.UK v2.0 2022** **Weakly encrypted traffic ORANGE** **Unencrypted traffic RED**	Many other lower bandwidth network services (such as database (SQL & Oracle), NTP, SMTP, SNMP, DNS, Snow, Ivanti, SMB (Isilon or other file shares) and others) may still be unencrypted in transit. Unencrypted network traffic can be easily intercepted and even modified, which could threaten the Confidentiality, Integrity and Availability (CIA) of the information. As a UK Government Arm's-Length Body (ALB) KEP and our suppliers are mandated to meet the Secure by Design (SbD) Principles (see Secure by Design Principles - UK Government Security) which recommends end-to-end encryption for all network traffic.
RR02	Thus far most of the deployment work in our Azure tenant has been done for us by consultants - e.g. Risual, the A4E supplier - so we will need to upskill the internal teams.	The exact impact is not yet known as we have yet to implement Fabric in our tenant, so it is unclear exactly what skills we'll need to deploy this architecture. This will be closely monitored by the project and has been raised with the Project and IT SMT. In addition the project will aim to follow industry standard guidelines such as the Microsoft Cloud Adoption Framework (CAF) and ensure all build activities are carried out by the KEP technical teams – with advice and guidance from contractors/third parties as required.
RR03	Cost control of running the new Fabric environments	Certain unknowns around how much data processing will cost over what is currently run using Business Objects
RR04	Resource availability – the Data Transformation project is a multi-year project that will be particularly impactive for the IT Data Engineering Team. A Change Freeze will have to be agreed with the business for the SAP Business Objects reporting landscape to enable the teams to fully focus on building and developing the new platform.	A Change Freeze has been requested and is in the process of being authorised.

7. Glossary

Term	Description
AD	Active Directory
ADF	Azure Data Factory
AIR	Azure Integration Runtime
A4E	Analytics For Everyone (an Essex-only early implementation of Power BI in the KEP tenant)
BRC	Back Record Conversion
Bicep	DevOps scripting language for deploying infrastructure
Business Objects Universe	Processed / curated data cluster that has already been through ETL, readied for direct connection from reports
CAF	Microsoft Cloud Adoption Framework
CI/CD	Continuous Integration/Continuous Deployment – a DevOps approach to data engineering
Data Cleansing	Updating data characters, data types and data structure to follow data standards that allow operations such as ingest to databases and association to structured data
Data Lake	Cloud data storage for efficient and low-cost structured and unstructured data
Data Lakehouse	Data warehouse using parquet files instead of database tables, on a data lake
Data Mart	Logically grouped data store, often referred to as a silo
Data Mastering	Refining multiple versions of the same data to a standard eg. Mr, Mr., mr, Mister all standardised to Mr
Data Warehouse	Separate database store from operational systems, optimised for reporting
Delta Check	Difference between the records in two tables, highlighting the difference
Delta Table	Abstraction layer over data lake data to capture changes. Format used in Fabric to allow updates to parquet files that are read-only natively
DevOps	Combination of Development and Operations that captures and facilitates creation and updating alongside, but without interfering with, live / production environments
Dimension Table	Data Warehouse table for descriptive data, mostly text
Entra	Microsoft's family of identity and network access products that enables implementation of a Zero Trust security strategy
EP	Essex Police
ETL / ELT	Extract Transform Load / Extract Load Transform - Methodology for curation of data
Fabric	Microsoft's SaaS based Azure tool for data operations and presentation
Fact Table	Data Warehouse table for aggregate-able data, mostly numeric
Git	Code repository integrated with DevOps
HA	High Availability - Azure backup and retrieval protocol
IDE	Integrated Development Environment - Visual Studio Code, Fabric Data Factory etc
IaC	Infrastructure as Code - creation of Fabric and Azure resources
IPsec	An encryption protocol
KEP	Kent Police and Essex Police
KP	Kent Police
MDF	Metadata Framework
Medallion Architecture	Best-practice methodology for structuring reporting and analytic data, and optimising ETL / ELT operations for Business Intelligence and Management Information reporting
Parquet File	Azure Data Lake file format, self-indexed for efficient retrieval and querying
PDS	Police Digital Services
Power BI Desktop	App for creating Power BI reports for publishing to Power BI Service
Power BI Report Builder	App for creating paginated reports - multiple pages like in MS Word, for publishing to Power BI Service

Power BI Service	Power BI cloud where reports and dashboards are accessed from
Repo	Repository in Git for storage of code artefacts
SaaS	Software as a Service
Sandbox	Secure enclave Azure subscription for use as a test / experimentation, separately from development, testing and live environments
SAP Data Services	Business Objects ETL Tool
SHIR	Self-Hosted Integration Runtime. Connects Azure to on-prem data
Structured Data	Data that has been formatted to database design standards
Truncate & Reload	Methodology for cyclic ingest of data where destination table is emptied prior to load of newly ingested data
TLS	Transport Layer Security
UAT	User Acceptance Testing
Unstructured Data	Data that requires transformation to relate with structured data
VM	Virtual Machine
YAML	DevOps pipeline deployment scripting language. Stands for ‘Yet Another Markup Language’
ZRS	Zone Redundant Storage - Azure backup protocol

8. APPENDICIES

8.1. Secure by Design (SbD) approach - UK Government Security

8.2. Secure by Design Principles - UK Government Security

8.3. Device Security Guidance - National Cyber Security Centre (NCSC)

8.4. Data Transformation - Testing Strategy

8.5. Naming Conventions

The naming conventions used follow current guidelines from internal naming policy documentation and PDS design. These are high level for VMs, servers, subscriptions and resource groups etc. The conventions have been extended by adopting recommendations from Microsoft, for resources that run within the resource groups, servers and VMs.

For Azure resources - Azure Data Factory, Fabric Workspaces etc. the following pattern has been used:

<resource abbreviation> - <usage persona> - <environment>

Resources at the next level in - Lakehouses, Pipelines, Notebooks etc. use the following pattern:

<RESOURCE ABBREVIATION>_<Resource_Identifier>

Resource Type	Name
Azure Data Factory	adf-mdf-dev adf-mdf-qa adf-mdf-uat adf-mdf-live adf-dataengineering-dev
Fabric Workspaces	ws-athena-dev ws-dataengineering-dev ws-datascience-dev
Purview	pview-dataengineering-dev
Lakehouses	LH_100_Bronze LH_200_Silver LH_300_Gold
Pipelines	PL_Metadata_Grandparent PL_Metadata_Parent PL_Metadata_Child
Notebooks	NB_05_Bail_Bronze
Dataframes	DF_Athena_Source
Variables	V_Crime_Ref
Parameters	P_Crime_Ref

If these are agreed when reviewed, they can be incorporated in the KEP naming document.