Data Mesh SaaS Implementation - Design and Delivery

GCLOUD 15

Release Date: 2026-01-30

Copyright Uncommon Correlation

Overview

Implement a data mesh in dynamically provisioned, vendor-agnostic, well-governed, hybrid clouds, using SaaS and open-source offerings. Achieve auditable, and sovereign, control over your domain data by implementing the four principles of the data mesh: ''domain-driven ownership of data'', ''data as a product'', ''self-serve data platform'' and ''federated computational governance''.

Approach

We begin all projects with domain-driven-design, and behaviour-driven-design, exercises, to build a representation of your target system or software, which we call ‘features’. From this, we produce a body of knowledge which captures your ambition, maps it to a strategy, and lays out a series of tactics to be embodied in your system.

The features, alongside other artifacts such as an ontology or a domain language glossary, are used as an ‘outside-in’ testing framework, which we encapsulate into our automated continuous delivery pipeline executable feature or function tests. We pair this with an ‘inside-out’ set of formal verifications and unit tests, which are also run in our automated continuous delivery pipeline. This ensures both the correctness and quality of our delivery, and customer satisfaction in the fulfilment of objectives.

During design and through delivery, we operate with a sincere focus on resilience and security. We have a robust and proactive risk discovery methodology that spans the compute stack from hardware to interface design, and the breadth of human factors such as supply chain attacks on software bills of materials.

Our clients delight in the usability and robustness of our systems.

Features and Benefits

- Cloud and SaaS vendor agnosticism: Protection against vendor lock-in
- Hybrid cloud SaaS deployment: Achieve best VFM for compute tasks
- Unconstrained flow of data: Standardise definitions of data in flight for agility and resilience
- True data sovereignty: Secure access to all data via RESTful API ensures ownership
- Authoritative data governance: Tamper-evident, distributed, recording of system activity for absolute integrity
- Distributed by default: Mitigate risk and evade attack by removing vulnerable centralization
- Rapid delivery: Surpass expectations at a fixed price
- Domain control: Encapsulate your domain in software to reduce processing problems

- Data as a primary concern: Better data and tooling to empower domain experts
- Domain evolution: Prevent spiralling costs when your domain or its data changes

What You Will Get

Design

- A domain model detailing the entities/ nouns and events/verbs of your use cases expressed in crows-foot or entity-relationship notation
- Features comparison of SaaS offerings with mappings to cloud vendors which support them in a standardised manner
- A set of example data definitions / schemata, expressed in documentation and a standard such as JSONSchema
- An ontology
- Use cases expressed as both user stories and gherkin scripts
- A domain language glossary of all concepts
- Formal verification scripts (where formal verification is required) in a suitable formal verification language (like LEAN or F*)
- Cloud architecture requirements assessment
- Software Bill of Materials (SBOM) projection
- Projected risk assessment

Delivery

- A system or software that meets your use cases
- How-to documentation for end users
- How-what-why documentation for developer users, covering architecture decisions, as well as implementation details
- Formal verification scripts (where formal verification is required) in a suitable formal verification language (like LEAN or F*)
- Cloud architecture diagrams
- Software Bill of Materials (SBOM)
- Risk assessment
- Ownership of all code and IP developed on contract

- Ownership of all background code and IP required for the above code and IP (where not Open Source licenced - for which you will receive all software bill of materials)
- Ownership of all data you load into the system or software
- Joint ownership of all data we load into the system or software to ready it for production
- Ownership of all data that is calculated from data in the system or software
- RESTful APIs to all data structures within the system

Pricing

This service is divided into two elements, with separate costings: design sprints and delivery projects.

Design

A design sprint is a two-week endeavour, costed at a baseline equivalent of three full-time-equivalent team members, spread across the following disciplines:

- Hardware and infrastructure
- Back-end development
- Front-end development
- Aesthetics and interface design
- Theory and semi-formal / formal methods
- Strategy and planning
- Data modelling

The design sprint will produce the materials detailed in the Design sub-section of the What You Will Get section, above.

Each sprint costs £90,000.00. Extra full-time-equivalent team members can be added for £30,000.00 per sprint. All team members are at SFIA level "Set strategy, inspire, mobilise".

Delivery

The delivery project costs four full-time-equivalent team members, per day, per feature, as drafted according to our standards as produced by our delivery sprint. Each feature is assessed against the SFIA level required, and costed according to our rate card.

For projects where feature requirements have already been set (by the client or by another partner), an extra analysis fee is charged at one full-time-equivalent team member, per feature, at SFIA level "Set strategy, inspire, mobilise".

For projects where the underlying cloud systems for the well-governed model have already been provisioned (by the client or by another partner), a familiarisation and configuration fee is

charged at a one full-time-equivalent team member, at the appropriate SFIA level for the feature.

The delivery project will produce the materials detailed in the Delivery sub-section of the What You Will Get section, above.