Published in Intuit Engineering

Tristan Baker Follow

Feb 17, 2021 · 14 min read · Listen

Save

Intuit's Data Mesh Strategy

Intuit's mission is to '*Power Prosperity Around the World*'. And Intuit's strategy for delivering on that mission is to be an '*AI-Driven Expert Platform*'. I work on Intuit's Data Platform team and we play a critical role in delivering the solutions required to make that happen. My team's internal customers include Intuit teams that build the products and offerings responsible for powering prosperity for Intuit's consumer, small business and self-employed customers.

My colleague, Mammad Zadeh, Intuit Vice President, Data Platform, recently published an article describing what we've been up to over the past couple of years. We've migrated from an on-premise architecture of centrally-managed analytics data sets and data infrastructure tools to a fully cloud-native and much more expansive set of data and tools th 616 8 ams to build a wide range of streaming, machine learning, and analytic processing workloads. Intuit's customers are reaping the benefits of this new architecture and I'm proud of what we've accomplished.

However, as far as we've come, the work is unfinished and opportunities abound.

We surveyed the landscape of problems within our solution and researched what solutions existed for companies with a similar size and scope as ours. Zhamak Dehghani's description of a "Data Mesh", which favors pushing data ownership and responsibility to local domain teams at the edges of the business, immediately caught our interest. The opening "Architectural Failure Modes" section is a frighteningly accurate summary of the problems that plague our current state. Zhamak's advocacy for Domain Driven Design, with an emphasis on ownership and accountability of singularly focused, problem domain-oriented teams, as the backbone of a strategy to address these issues, resonates loudly. We've seen this approach work well at the "front of the house" with our microservices architecture and development of Intuit's core transactional services. So we were confident that applying that strategy to our "back of the house" data systems could be equally successful.

The rest of this article will take you through a full articulation of our vision, inherent challenges, and strategy for building better data-driven systems at Intuit. You can think of it as data mesh, interpreted through an Intuit lens. If you've spent any time in a typical data lake and caught yourself wondering, "How the heck can we decrease pandemonium and increase productivity to get back to the business of making customers happy?!", read on.

Vision

Intuit needs data-driven systems to enable smarter product experiences, and to enable more Intuit teams to more easily create them. This includes a variety of data workers:

- A service engineer building a service that publishes an event when an invoice is paid.
- A UX developer building a landing page that publishes an event when a user visits.

- A business analyst building a report that shows QuickBooks hour-by-hour new subscriptions counts.
- A data scientist building a model that predicts fraud before approving payment.
- A data engineer or analyst building a pipeline to identify new users and returning users by synthesizing data from our customer identity and Intuit back-office systems.

Challenges

To understand how to drive simpler, rapid development of these systems, we first had to discover what was inhibiting the data workers trying to build such systems. We surveyed 245 current users of our data systems to find out what kinds of questions they were struggling to answer in the course of their daily activities. Across the roles of business analysts, data engineers, data scientist, and machine learning engineers, these questions bubbled to the top:

Data Discoverability

- Where can I find data about a particular thing (customer, company, etc)?
- Where can I find the data sourced from a particular product or service?

Data Understandability

- Who can approve my access so that I can see samples of the data?
- What is the schema of the data?
- What is the business meaning and context of the data?
- Is this data related to other concepts? Is it joinable to other data? What is the meaning of the relationship?

Data Trust

- What system produces this data and at what latency?
- What other systems use this data?

- What is the quality of this data?
- Which team supports this data if it breaks?

Data Consumption

- How is this table/topic partitioned?
- Who can approve my production system to access it?
- Will I be alerted if the schema changes?

Data Publication

- How do I describe my data so that others understand what it means and how to use it?
- Where do I host my data so that other systems can access it?
- Data systems are complicated; how can I build and operate one?
- What are my operational responsibilities once my process/data is in production?
- How do I meet my compliance requirements for processing, storing, and publishing data?
- Am I duplicating processing/data that already exists?

Strategy

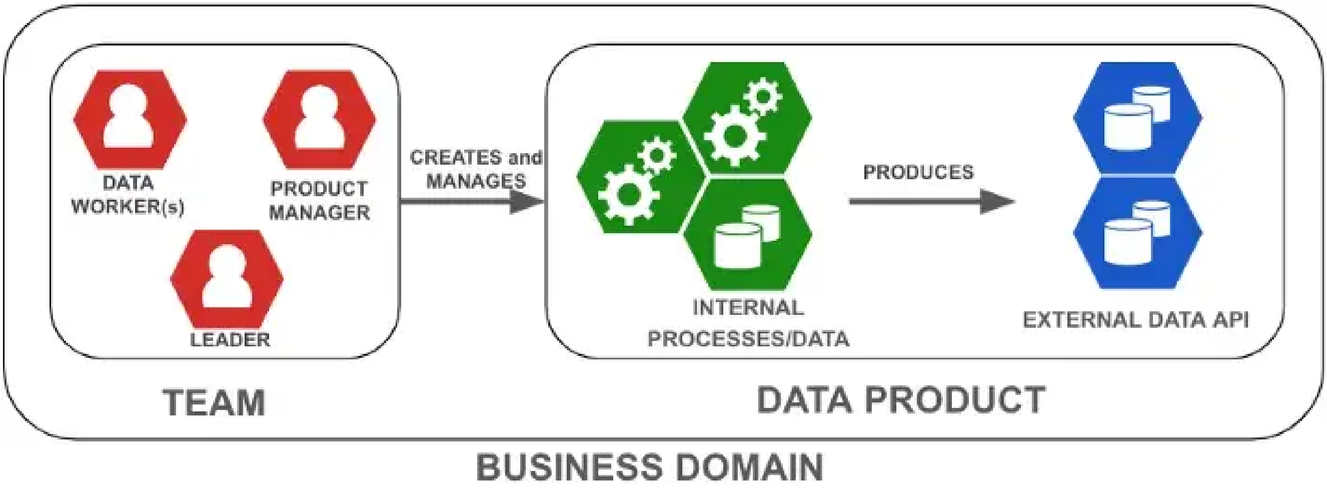
With new insight into our users' experience, we defined a strategy that will enable a greater number of data workers to easily create and own high-quality data-driven systems; resulting in smarter product offerings to Intuit customers. The strategy empowers data workers to design, develop, fully describe and actively support their own data driven systems, allowing the next person to easily discover, understand, trust, and consume data. This powers the network effect of data production and consumption, which increases the productivity of all data workers, resulting in the large variety of high quality data driven systems Intuit requires.

Discovery, understanding, trust, consumption, and publishing of data systems is supported in three specific ways:

1. A systematic approach to organizing the people, code and data which in turn map a solution to a business problem and its owners.
2. Ownership that ensures all teams understand and are accountable for a set of defined responsibilities in building and managing their solutions; including adherence to a set of defined best practices to produce only high quality data.
3. A rich suite of products that enable teams to more easily author, deploy, and support their own solutions, and to do so according to best practices.

Organizing People, Code and Data

To more precisely describe the set of related code and data that collectively solve a business problem, we introduce a systematic method of organizing them: **Data Product.**



A Data Product is a set of internal processes and data that produce a set of externally consumable data, all aligned around the same problem in the business domain. It is created and managed by a team collectively capable of understanding both the business problem and the implementation details of the internal processes and data that solve that problem.

This is in contrast to the current state in which teams are aligned around technology, forcing the same logical business problem to be arbitrarily divided among several different teams. This creates inefficiencies in communication and coordination that cause many of the issues teams experience today. A team is likely composed of one or more Data Workers versed in the technologies used in the solution, a product manager versed in the business problem, and a leader responsible for the timely and efficient execution of the project that delivers the solution to the business problem.

It is critically important to develop a system for organizing these assets so that we can layer the concept of ownership on top. Without the ability to pinpoint exactly which humans are responsible for what code and data, there is no ability to hold anyone accountable for making sure that it is of value to the company, and thus no reasonable path towards making things better.

To zoom in on just Intuit's data lake environment for a moment, we haven't had a full understanding of the purpose of an overwhelming majority of hundreds of thousands of fields across the thousands of tables it contains. This cruft has meant that nearly half of data workers' time in the data lake has been wasted due to an inability to find data or an owner of that data to explain what it means.

Ownership

Building on this system of organization, ownership brings together three important concepts:

1. An owning team,
2. A scope over which that ownership applies, and
3. A set of responsibilities to fulfill within that scope.

Everywhere there is a Data Product, we require a clear statement to the following effect: "Team Awesome (i.e., owning team) is responsible for the authorship, rich description, governance, quality and operational health (i.e., responsibilities) of the Fantastic Data Product, which is composed of these processes and data (i.e., scope)".

Team and scope are already covered in the definition of Data Product earlier in this article. Responsibilities are defined below:

Authorship — The owning team designs the solution, authors the code, and arranges the systems that implement it

Description — The owning team describes the system in enough detail such that other teams are able to discover, understand and use the Data API.

Governance — The owning team is responsible for the solution adhering to all applicable compliance and security requirements.

Quality — The owning team is responsible for the quality of the solution and delivering its data API with the promised correctness and latency.

Operational Health — The owning team is responsible for the Data Product's uptime, performance and cost.

Implicit in the listing of the above responsibilities is that we require teams to be excellent in fulfilling them. A bad Data Product (e.g., one that always breaks, isn't well described and for which no one can find the owner), does more harm than good. Therefore, we require *well* designed, *richly* described, *strongly* governed, *high* quality and operationally *reliable* Data Products.

To help teams understand these expectations, we've established a framework to enumerate specific criteria that must be met in order to be considered high enough value to the business to be published. The framework describes two levels, and the various criteria required to meet those levels:

BASIC describes the minimum set of criteria that must be met to comply with our system of organization and ownership responsibilities.

BEST goes beyond BASIC to introduce the notion of rationalized data that ensures a Data Product is producing content that is of net-new value to the ecosystem and does not duplicate or contradict important concepts that have already been established and relied upon across the company. This is getting into the world of master data management, which fundamentally pits the goals of a system of federated ownership responsibility against the goals of centralized data definition (how we plan to do this elegantly and at scale at Intuit will be the subject of another post).

A BASIC Data Product has —

- Direct upstream dependencies identified
- Compliance regulations identified
- Change control and access control processes in place
- Data API decoupled from internal implementation
- Data API schema defined and described
- Data API monitored against stated SLA
- Owning team is assigned, accountable for and measured against their responsibilities

A BEST Data Product has —

- All of the BASIC stuff, plus
- Data rationalized against and related to all other Data Products

A Diverse set of Composable Capabilities Enabling Best Practices

A diverse set of composable capabilities gives teams the tools to design, author, deploy and operate Data Products for themselves. The diversity ensures that the equally diverse skills and perspectives of data workers are accommodated and that everyone can be productive in consuming and producing Data Products. The built-in adherence to best practices ensures Data Products are constructed and data is produced in a way that advances, rather than erodes, Intuit's ability to leverage all of the data at our collective disposal. Each offering in the set enforces that its users produce Data Products that meet the BASIC (or better) requirements for publication.

Each of these capabilities is integrated with central platform services and standards that ensure that security, authentication, authorization, access control policy management, change control procedures, schema registration and documentation are implemented in a self serve manner so that individual teams are able to efficiently fulfill their ownership responsibilities. Additionally, quality, cost, and performance metrics are measured and reported in the same way, so that the individual teams — as well as the organization at large — have insight into how well (or not) a Data Product is operating.

The listing and brief description of all product offerings that will be available to Data Workers is as follows. Since many of these products are big and complex and well established in their own rights, I've provided only brief summaries. Where possible, I've linked out to other Intuit published articles with more information.

Capabilities Used by Teams to Implement Their Data Product

Instrumentation Services — A solution for instrumenting user facing experiences with behavior tracking events in compliance with Intuit's event standards so that a UX Team can track user interactions and so that other Analyst and ML Teams can consume these events from well defined locations for further analysis and transformation.

DomainEvent Services — DomainEvents describe the changing state of the sending system. The DomainEvent Services are leveraged by Service Engineers in their transactional microservices to provide DomainEvents asynchronously to downstream Data Products. This is intended to be used in place of where raw CDC data is used today.

Universal Ingestion Platform (UIP) — Leverages the CDC streams and/or jdbc connections supported widely by transactional database systems (Oracle, MySql, etc) to pull data into the Data Lake so that it can be accessed by big data processing technologies. UIP is leveraged by Service Engineers to make raw data available to the Data Lake only when DomainEvents are not feasible. A platform constructed from Oracle GoldenGate, Debezium and DeltaLake.

Intuit Persistence Services (IPS) — Enables Service Teams to deploy and manage databases in support of stateful business logic. The deployments encapsulate all Intuit best practices and standards to enable highly available and secure systems.

Stream Processing Platform (SPP) — Enables Service Teams or Data Engineering Teams to create systems that consume, process and produce streams of real time asynchronous events. Detailed articles here and here.

Pipelines — Enables Data Engineering and Analysts teams to create spark and hive batch processing jobs and schedule them for execution at periodic intervals or upon the availability of results from other upstream Data Products. Detailed in an article here (the article is a bit dated now and some things have changed, but it should give you a general idea)

Schema Manager — Enables UX, Service, Data Engineer, and Analytics Teams to create the schemas that their DataProducts produce. The Schema Manager is able to enforce schema standards and recommend best practices as well as detect and prevent duplicate and conflicting definitions among data from different teams.

Entity Builder and Entity Resolver — Enables Analysts and Data Stewards to define logic that curates and masters entities so that cross cutting concepts (like Customer, Product, etc) have consistent definitions available for use across the company. *We've started defining and building this; so it will likely be the subject of further blog posts.*

ML Platform Feature Engineering — Enables Data Scientists and ML Engineers to define features to be used in model training. More information about the feature, training and execution capabilities of MLPlatform can be found here.

ML Platform Training Service — Enables Data Scientists and ML Engineers to train models.

ML Platform Model Execution Service — Enables Data Scientists and ML Engineers to deploy trained models to be executed in real time or batch.

Data Quality Services — A capability available wherever a Data Product produces data. It assists all users by providing context specific quality checks on their data. It supports quality judgement during the build and testing phases as well as in-production quality observation and monitoring to ensure data quality targets and data SLAs are met throughout the data lifecycle. We're using Monte Carlo for this, and will be supplementing it with some of our own innovations, too.

Data Discovery and Exploration (DDE) — A tool that allows a user to browse available data sets and fire up a query browser or a notebook to start accessing and manipulating data sets. Assists all users in the R&D work that usually precedes any effort to build and release a Data Product into production.

Capabilities Used by Teams to Host their Data Product's DataAPI

Data Lake — The Data Lake hosts large scale data for analytics, machine learning and other scan-and-process-in-parallel-a-lot-of-data-as-fast-as-possible workloads. Leverages HMS, parquet and S3 blob storage. It establishes the authorization model and workflows that enables consumers to request and producers to approve access to data sets for both exploration and production access purposes. More detail here and here.

EventBus — Location for all streaming event data. Directly leveraged by Service engineers and Data Engineers implementing asynchronous eventing solutions or creating streaming processing applications. Indirectly leveraged by Analysts, Data Scientists and ML Engineers when they leverage the real time processing capabilities of ML Platform Feature Management and the Entity Build and Resolver. More detail here.

Data Mart — Location for data used to drive business reporting. Primarily used by Data Engineers and Analysts. Data Marts are purpose built schemas optimized for particular areas of the business (e.g., "tax expert case management and efficiency analytics"). They provide both the data sets and the query engines required to drive Dashboards.

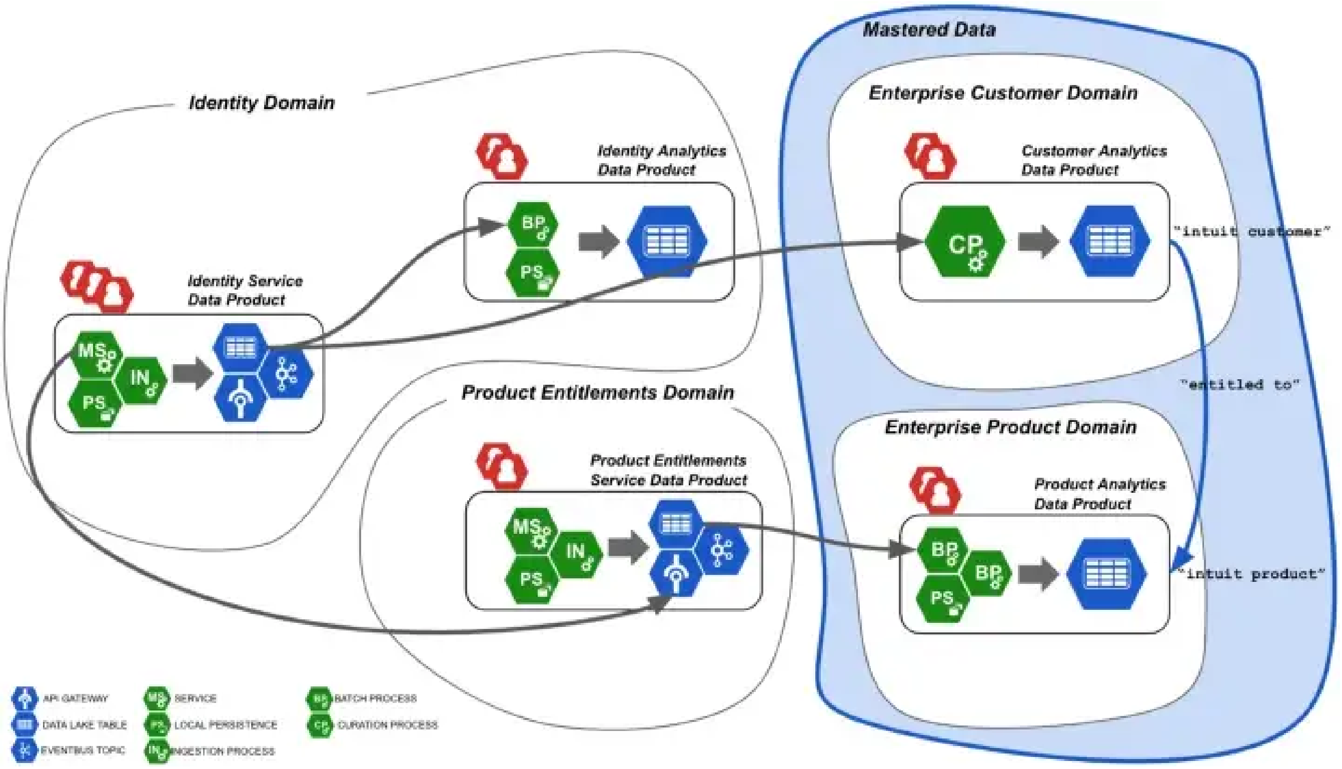
FeatureStore — Location for all features required to drive ML training and prediction use cases. It allows the same feature to be hosted in an online and offline location for real time and batch access while maintaining equivalence between the two — this equivalence guarantees stability between the data the model was trained on and the data the trained model uses to generate predictions. Leveraged by Data Scientists and ML Engineers.

API Gateway — A location for all services to host their REST APIs for exposing service behavior (ie, "POST /TaxService/FileTaxReturn/{id}") and data (ie, "GET /TaxService/TaxReturn/{id}"). The gateway handles authn/authz in a standard way across the entire enterprise, thereby ensuring consistently applied security across all services in the organization.

For Example

The collection of all meta-data that describes all Data Products is used to create a universal metadata registry (ours is built on Apache Atlas) of all the teams, systems, data, entities, and the dependency relationships between them. It is the foundation for answering all questions about Data Product ownership, scope, responsibility, technical dependency and data meaning.

A collection of five imagined Data Products, their technical dependencies (grey lines) and their rationalized data definitions and relationships (blue lines) are drawn below:


Mastered Data
Identity Domain
Identity Analytics Data Product
Identity Service Data Product
Enterprise Customer Domain
Customer Analytics Data Product
"intuit customer"
"entitled to"
Product Entitlements Domain
Product Entitlements Service Data Product
Enterprise Product Domain
Product Analytics Data Product
"intuit product"
MS
IN
PS
BP
CP
API GATEWAY
DATA LAKE TABLE
EVENTBUS TOPIC
SERVICE
LOCAL PERSISTENCE
INGESTION PROCESS
BATCH PROCESS
CURATION PROCESS

Note that questions that we led with at the beginning of this article and drive our strategy are answerable:

- Where is Customer data available for consumption? What is the schema of the data?
- Which other Data Product(s) did Customer Analytics Data Product get its data from?
- Which team is responsible for Product Entitlements Service Data Product? Who is on that team and how can I get in touch with them?
- How are Intuit Customers related to Intuit Products?
- Which team can approve my request to access the Customer Analytics Data Product's Data Lake table?

Up Next

Intuit is at the very beginning of a data mesh journey. Given that it's a relatively recent concept, no one has developed anything resembling a data mesh standard yet. While many of the capabilities described throughout this article exist in one form or another, the systems required to organize them are just getting underway.

This article is a lot about the why and the what, I expect my next post to get into more detail about the how. My hope is that this article, and any that follow, might provide other like-minded state-of-data malcontents with a running start in their own endeavors. As this concept gets more mature and gains momentum, I'm interested in hearing your ideas for how we might standardize on some of these approaches. A collective aligned effort might speed things up a bit for all of us. If you are one of those people, please get in touch.

Technology Data Data Mesh Big Data