Daml Hub Product Knowledge Base

Use Cases

Daml and Daml Hub combine to create a framework designed for building composable, flexible applications in what was otherwise a stodgy, slow category. Only write the code that describes the behavior of your application, never to manage any of the infrastructures. When deployed in Daml Hub, you will pay for what you use at the ledger update level instead of overprovisioning and underutilizing cloud infrastructure.

Autoscaling application backends

Complete Daml applications can be deployed without any infrastructure configuration or engineering effort. Your full application can be built from scratch and launched to production in a matter of days. Applications running in Daml Hub scale as your traffic increases and you'll only ever pay according to your activity.

Multiple-Page Application (MPA) backends

Daml application backends easily support the functionality required to drive feature-rich, multi-page applications. A single file can be responsible for and generate all your APIs, roles, access controls, permissions, and everything you need to drive a complex application. Daml decouples each step of a workflow by breaking it down into a series of functions and state changes. Your Daml file also describes highly granular controls over data permissions across a complete workflow. Daml Hub simplifies authentication and authorization schemes by mapping an authentication layer to your Daml definitions.

Event-driven applications

Daml applications produce tangible and reproducible event-driven applications by capturing the sequence of events as they occur in several disparate systems and making that sequence available to create actionable conclusions. When deployed in Daml Hub, that facility is made available at a low cost. In addition, Daml applications integrate well with external systems and stateless components, creating a scalable event-driven application across all components.

Workflow engines

Many businesses - particularly those at the beginning of a digital transformation - seek to consolidate, encode, and refine their core business processes. This can be done by capturing those workflows within Daml's declarative and self-descriptive language, which describes the separation of actions according to the role of the actor and combines those actions within a broader state machine. Daml is an ideal language for capturing and extending this state machine design, and with Daml Hub you can bring that state machine to life in just a few clicks, over the internet!

Transparent and auditable applications

SaaS applications are increasingly coming under scrutiny about the data held by the application and what other services the data has been shared with. Users of applications running in Daml Hub can view all the core data within an application that is relevant to their activity. In addition, users can audit which entities are entitled to access the data and have active access to the data at any given time. Building applications in this way allows developers to build and gain trust with their users over time, increasing the number of services and information the user will be willing to share with the application.

Comparisons

Daml Hub has elements found across PaaS, iPaaS, FaaS, and SaaS platforms. Applications running in Daml Hub benefit from having a managed event-driven system with built-in authorization and API management. You can also integrate and call external APIs within your application by adding the API call to your application workflows. To deploy your application, all you have to do is push your compiled code to the service. We take care of executing and running all the components for you - from the UI down to the core backend. All applications deployed in Daml Hub come with an out-of-the-box UI that allows you to test and demo your application even if you have only built your backend code. You can even begin to engage your users through the Daml Hub UI as a way to get very early feedback while you think through the UX and consider the features your users need.

IaaS vs Daml Hub

Infrastructure as a Service (like AWS, GCP, and Azure) providers give users the ability to manage a virtual set of components traditionally found in a data center. IaaS is largely unopinionated - developers can build just about anything that they need and run it in IaaS. However, developers are responsible for building all the code that manages both the infrastructure and all the components of the application. Often when provisioning IaaS, organizations over-provision infrastructure that they may not use when idle or struggle to burst when activity peaks. Managing applications running in IaaS often requires skilled DevOps engineers to work side by side with core application engineers to ensure all components are built to work together. Daml is a full application development framework that allows developers to focus solely on describing the business logic and behavior of their applications. Daml Hub is an opinionated environment designed specifically for Daml applications. When used together, a developer can build and deploy a complete application backend from a single Daml file. The Daml framework and Daml Hub environment will use that file to generate everything you need including database schema, APIs, and managed authenticated access to the application. Managing a Daml application in Daml Hub is reduced to turning components on and off, deploying new code and integrations, and upgrading features as you iterate over time. Your team will never need to wrestle with complicated DevOps requirements or infrastructure concerns.

PaaS vs Daml Hub

Platform as a Service is another level up of abstraction over IaaS for the developer. Most PaaS platforms offer the next layer of commonly required components like operating systems, databases, and a set of developer tools. PaaS platforms tend to be more opinionated than the IaaS layer since they are made to aid developers in application development - sometimes very specific types of applications or problems. Developers using PaaS are writing less code and have fewer DevOps requirements than those using IaaS but they still have plenty to account for. Applications built for PaaS still have to account for scaling their application across servers, components, and many other components. To help alleviate, there is often a wealth of modules and integrations that are built to run side by side with the developer's application in the PaaS environment. Daml Hub aspires to follow in the early footsteps of Heroku - which at the time was a purpose-built cloud experience around the Ruby language. In many ways, Heroku helped complete the value prop of Ruby and jointly created a great web development experience for developers. Applications built for Daml Hub have a higher level of abstraction as the developer is completely unaware of the underlying infrastructure and many of the key modules required for a developer's application are directly built into the service (eg authorization).

iPaaS vs Daml Hub

Integration Platform as a Service is a fast-growing category of cloud service purpose-built for building and deploying integrations in the cloud. Notably, these services are often used to help connect legacy enterprise applications to key cloud services like Salesforce. iPaaS services include tools for helping users build custom integrations as well as use those made by other users of the platform. iPaaS is typically used to help bridge the gap between old and new technologies across a variety of deployment footprints which is why it is sometimes thought of as a hybrid cloud solution purpose-built to help make on-premise

systems accessible in the cloud. Integrating with 3rd party and legacy APIs is a critical component of any successful new application. However, when integrating with many APIs, it can become difficult to manage and maintain over the lifetime of your application. Daml Hub makes it possible to include calling external APIs in a simple Daml application call without having to build an extra normalization or aggregation layer for your integrations. This means adding new APIs to your application is as easy as writing them into your core workflows without having to learn how to add and support them from your infrastructure.

FaaS vs Daml Hub

Function as a Service (or sometimes referred to as a serverless) is a model where the cloud provider is responsible for running and executing your code. FaaS has traditionally been used for edge computing services and processes that lend themselves well to simple functions and straightforward transformations. Serverless applications are typically stateless and are connected to some external persistent store or event-driven system. The serverless experience has been a strong inspiration to the Daml Hub team. We appreciate the simplicity of thinking about processes as a series of discrete functions and state changes. We are also big fans of empowering developers to focus on the code that differentiates their applications and not getting buried in the weeds of getting your code to run. Daml Hub is a serverless experience for more stateful applications. This means you can build the next eBay or Airbnb with a serverless experience from top to bottom without any external persistent store or extra components. Applications running in Daml Hub benefit from connecting to serverless or function-as-a-service systems to enhance the functionality and automation of the application running in Daml Hub. A few of our early users have even integrated their Daml Hub application into AWS Lambdas with great results.

Example Application Ideas

All the application ideas presented below are ideas that we have discussed openly or built ourselves. Each example lends itself well to a Daml application running in Daml Hub because of the ambitious nature of the workflows, the inherent key interactions with users of the applications, and the benefits to everyone from a more transparent view into how the workflows and data are being shared with others. We would be happy to talk to anyone in greater depth about these ideas and how we could help you launch the next great application. We have direct expertise with all these types of applications and have the means to help you get your idea off the ground quickly. Don't see your idea on the list? Reach out and ask us about your use case.

2 Sided Markets

Application network effects and 2-sided market-style applications can build robust communities of like-minded users at opposite ends of a supply chain. Any gig, share, or circular economy application can benefit from the built-in properties of Daml applications that were designed to make building some of the world's largest markets a breeze.

Example : Apartment swapping during COVID

During the Covid-19 lockdowns, AptSwap built and launched a platform for people to swap apartments for extended periods. The platform was quick to launch in the middle of the lockdowns and provided people within and across cities to find safe respite, comfortable ways to change their living arrangements and ease the burden of lockdown.

Example : Community bartering

As COVID has slowed down travel and broader global interactions, CommBar Inc launched a simple way for neighbors to connect and offer ways to help and support outside of unemotional economic help. CommBar setup a simple marketplace for people to connect, request, and offer different skills and services to their neighbors as a way to help their local community thrive during the lockdowns. CommBar Inc was able to crowdsource building their open-source application. Daml made working on an open-source application easy to do across people who had never worked before thanks to the ease of reading and understanding the intent of the application code. Daml Hub reduced the number of items the community had to build and removed any need for them to build and deploy their infrastructure for the application.

Digital Exchanges

The world of digital exchanges is evolving quickly - from the concept of trading collectible sneakers to offering competitive upstart securities exchanges - and are changing the way all sorts of products are sold in the market. Daml comes with open source libraries that make it trivially easy to build a new type of exchange. Better yet, all other exchanges using these libraries are immediately interoperable with each other thanks to common data and workflow modeling. This allows the developer to focus on the key differentiating aspects of the exchange and not the minutiae of how to shape a bond or sneaker in financial terms.

Example : Social Networking Finance

CrowdFin Inc recognized that access to many private markets was being artificially constrained by asymmetric information and deal flow. To open up a wide variety of opaque private markets, CrowdFin Inc deployed a new model for investing. On their Daml-empowered platform, they allow any type of private investment to be structured where a pool of underwriters, entities that can perform due diligence, and other specialized functions must openly post their historical records. Investors can choose to follow other investors or follow specific key enablers for deals getting created. Following investors and enablers into deals gives a small percent of the earnings to the leads and all returns on the platform are public to the investors.

Example : Private market rebalancing markets

Venture capital and public equity funds provide key liquidity to companies at particularly risky moments in their lifecycles. As those companies mature and the portfolios of the VC and PE companies evolve, there comes a strong need to rebalance

exposure to certain sectors and types of investments over time. Liquidity for these portfolios can be few and far between so LiqVC.io built a marketplace where VCs, PEs, and accredited investors can offer exposure to slices of their portfolios to other investors in the network. The combination of a clever legal solution and strong workflow guarantees brought by their application brought a new pool of liquidity and flow to otherwise low-velocity portfolios.

Example : Institutional crypto wallet with built-in OTC capabilities

CryptoConnect, the institutional trading desk for cryptocurrencies, used Daml to build a fully-integrated wallet that has a complex workflow across their front office, middle office, back office, and IT departments. CryptoConnect realized their largest counterparties would benefit both from having their instance of the wallet and in the wallets being interoperable. They demonstrate the value of the wallets being connected by offering a blind over-the-counter market that can request a price, place an order, match, clear, and settle transactions seamlessly with the integrated wallet. CryptoConnect used the existing Daml Hub key management integrations to offer a robust and secure institutional wallet.

IoT and Data Marketplaces

As applications and networks grow, developers are in an increasing need for ways to make the data captured by these devices actionable. While many are dumping data into large data processing facilities to gain insight and find patterns, others are beginning to create open marketplaces for data.

Example : IoT Appliance Network

IoT Fridge Corp has been hard at work building a mesh network of sensors that let restaurant owners know when one of their walk-in coolers or freezers is having an issue. IoT Fridge Corp has created an advantage over its competitors by building a simple workflow where the IoT sensor records the event, checks local vendors for availability and quotes, and returns an event to the restaurant owner that allows them to not only know there is an issue but immediately schedule the remediation.

Example : Application Data Marketplace

New Freshman Corp has built an application that allows its users to coordinate and book services necessary to help incoming college students arrange for moving services and other key needs as they prepare to attend school. New Freshman LLC is often notified of the upcoming move months before any other service would be aware of the users' upcoming needs. This puts New Freshman LLC in a position to monetize these highly qualified sales leads that few others have. To do so, New Freshman Corp has built an auction exchange for their leads. The vendors have an opportunity to see a subset of the information about a user, place a bid for exclusive first access to the user, and be notified of winning bids and corresponding information.

Virtual games of skill and chance

Let's be honest - you didn't think we would mention virtual board games side-by-side with some of these other use cases. Daml is well-suited for building the rules for any online game where all participants care about the fairness and correctness of the outcomes of the game. Daml is better than your standard language for building these types of games because you can open-source the codification of the rules of the game while allowing each player to maintain proprietary strategies. Anyone can audit the rules and they are easy to read, modify, and extend in a transparent way to all other players.

Example : Fog of war chess

Everyone has had some exposure to the game of chess in their time but few have ever had to play chess with imperfect information. The challenge was to build a version of chess where each player can only see the pieces that it can attack as if in a fog of war situation. Daml's built-in privacy provisions made this a fun exercise for a new DA joiner to build as an onboarding experiment.