Digital Asset

Digital Asset Security Posture

Secure Products, Secure Data

Digital Asset

March 2021

Executive Summary

Digital Asset understands and appreciates the importance of security to our clients, which is reflected in how we architect, design, develop, build and distribute our products, and in how we protect our staff, our locations, our and our client's confidential data and services, and our infrastructure. This position paper describes how we view security and the implementation of controls to ensure this is enforced.

Table of Contents

1 Digital Asset Security Initiatives and Programs

1.1 The Digital Asset Information Security Program

The goals of the Digital Asset Security Program include:

- Protect the information, and privacy of our clients and partners.
- Protect Digital Asset staff, locations, data, services, and infrastructure from compromise or misuse
- Ensure the appropriate security capabilities (Confidentiality, Integrity, and Availability) are built into Digital Asset products & services
- Implement a secure Software Development Life Cycle (SDLC) process (plan, design, develop, package, distribute) to produce hardened, well-tested, high-quality products commensurate with their expected applications
- Ensure Digital Asset clients, partners, and operations teams deploy and run the product securely

1.2 Security Program and Governance

The Digital Asset Information Security Program covers two primary views of Information Security:

- Corporate Security: the security of Digital Asset people, locations, data, systems, and services
- Product & Service Security: the security of Digital Asset products and how we develop and deliver these to our clients

The Digital Asset Information Security Framework is diagrammed in Figure 1.

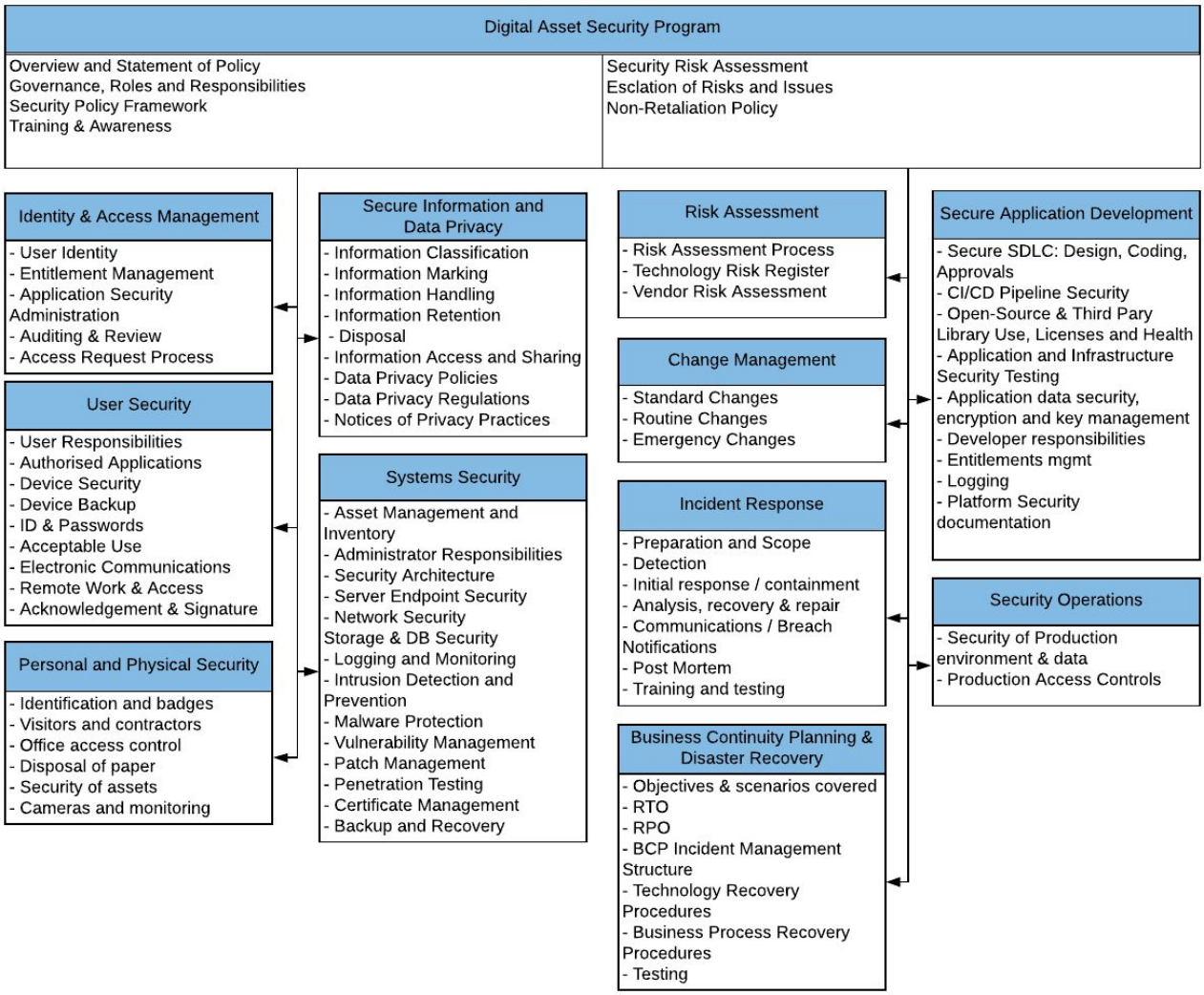

Figure 1: Digital Asset Information Security Framework

1.3 Digital Asset Security Team

The Security Team consists of staff who have made careers working for the largest Financial Services organizations. The team hold over 60 industry security certifications — CISSP, CCSP, CIPP, CIPM, ISC-CERT, CISM, CCSK, ITIL V3, and others —and has a breadth of experience across a wide variety of technology platforms and services.

1.4 Digital Asset Security Certifications

Digital Asset monitors a variety of initiatives and regulations to ensure that Digital Asset meets all regulatory, contractual, and compliance requirements.

1.4.1 SOC 2

Digital Asset has successfully completed the SOC 2 Type 2 independent audit with an unqualified opinion from our auditor. The scope of the assessment is our Software Development Build Pipeline with respect to the Security Trust Principle. This provides confidence and external validation that Digital Asset's software is protected against unauthorized modification or disclosure.

As the scope and range of services changes, we will also consider assessment against a broader scope of services and other Trust Principles.

1.4.2 GDPR

Digital Asset is compliant with the European Union General Data Protection Regulation (GDPR). We take the security and privacy of personal data seriously. Our privacy policy is published on our public website:

https://digitalasset.com/privacy-policy

Questions relating to our Data Privacy Policies can be directed to privacy@digitalasset.com

1.4.3 Cloud Security Alliance | Star Registry Participant

Digital Asset have completed and registered their responses to the CSA CAIQ v3.1 in the Cloud Security Alliance STAR registry. A version of that same response including all detailed comments is available under NDA from Digital Asset directly.

1.4.4 ISO 27001

Digital Asset is currently preparing to attest to ISO 27001/2 Certification. This is targeted for completion in 2021/22.

1.5 Responsible Disclosure and Data Privacy

Digital Asset maintains a Responsible Disclosure policy for reporting security vulnerabilities and concerns. The policy is detailed on our public website:

https://digitalasset.com/security/

The Digital Asset Security Team can be contacted at security@digitalasset.com

Questions relating to GDPR and our Data Privacy Policies can be directed to privacy@digitalasset.com

2 Security of the Company

2.1 Risk Management and Governance

Risk Management for Digital Asset is executed through the Risk Review Committee, with a membership consisting of the COO, CSO, CTO, and General Counsel. The committee meets monthly to review identified risks and prioritize mitigation efforts. The committee reports to the CEO and, ultimately, to the Board on the critical concerns of the firm.

2.1.1 Risk Management Policies and Procedures

Digital Asset maintains a set of industry standard policy and procedures, including:

- Information Security Program & Policy Framework
- Data Classification, Protection, Retention and Disposal
- Change Management
- Incident Management
- BCP and Crisis Management
- Vulnerability Management
- Vendor Risk Management

These are reviewed annually.

2.2 Data Governance and Oversight

All Digital Asset staff are required to review and acknowledge the Digital Asset Secure Information Policy and Data Privacy. This defines Data Classification tiers for all data and defines the appropriate handling, backup, retention, and destruction requirements.

Digital Asset requires appropriate risk assessment of all data services used for storage and processing of Digital Asset data, including data shared from partners and clients. Reviews are particularly focused on Personally Identifiable and Personal Health information, but also on the broader compliance requirements for GDPR and related data privacy regulations.

Digital Asset Risk Reviews are performed on all vendors/business associates. The use case, scope and data class in use drives the risk assessment process. Overall risk rating and general risk maturity are created specific to the vendor and the data class being handled.

2.3 Email & Web Security

Digital Asset has implemented mail gateway services for inbound and outbound email security. This filters and protects emails to staff, including spam, malicious emails, attachment and URL security, and malware protection. Phishing awareness training is provided to all staff and periodic phishing testing is performed to drive further awareness.

Digital Asset has implemented local endpoint agent and browser extensions to protect against website based attacks and other forms of spam and phishing attacks.

2.4 Endpoint Security

All users are required to use Digital Asset managed endpoints for work-related activities. All endpoints are managed centrally by MDM solutions, either JAMF Pro for Apple Macs or SaltStack for Ubuntu-based Laptops. Native capabilities of the device are leveraged for full disk encryption, firewall, and screensaver locks, which are enforced through MDM.

Digital Asset leverages a number of host-based agents for malware detection and prevention, data egress restrictions, audit, and oversight, including CrowdStrike Falcon Prevent for Endpoint Threat Detection and Prevention, CoSoSys Endpoint Protector for Data Loss Prevention (DLP), and Qualys for vulnerability scanning.

2.5 LAN Network Infrastructure

Digital Asset corporate network is in place between Digital Asset offices and cloud-provider private networks. Firewalls are used at all office locations to block ingress malicious traffic. Security Groups and firewall rules are used in Amazon Web Services (AWS), Google Cloud Platform (GCP) and Microsoft Azure to block ingress malicious traffic.

The network infrastructure is patched in a timely fashion, with monthly configuration samples taken as part of our audit-control process. Network traffic is sampled and reviewed, and flow traffic is sent to our SIEM for malicious traffic analysis.

Secured, full tunnel VPN is installed and available on all Digital Asset endpoints through a connection icon in the menu bar of each endpoint, should the Digital Asset employee find themselves working on an untrusted network.

2.6 Cloud Services and Infrastructure

Digital Asset leverages many cloud based services and also uses AWS, GCP and Azure to host and run our corporate infrastructure.

A mixture of public and private network segments, with firewall rules and access control lists in each cloud provider, are used to ensure only trusted entities are allowed access to resources within these cloud environments.

IAM roles and policies are in place to control user and administrative access to the cloud environments and administrative consoles. The principle of Least Privileged Access is followed when provisioning user and service account access to the cloud resources.

All rights and account access are regularly reviewed and inventoried as part of the Digital Asset audit process.

Security events from syslogs, network flow logs, and CloudTrail and Stackdriver logs are forwarded to SumoLogic's SIEM tool for centralized logging, alerting, threat detection and ticketing for remediation. The security team monitors the logs to identify potential threats and unauthorized activity and follows appropriate steps to remediation.

The Qualys Vulnerability Assessment platform is used to identify and remediate vulnerabilities found in all environments. Additional security tools such as Security Monkey, ScoutSuite, Forseti, and in-house developed solutions are used to audit and validate security configuration and access.

2.7 Personnel Security

2.7.1 Staff Background Checks

All Digital Asset staff are required to pass background and screening checks. This includes ten-year criminal background checks. These checks are performed at a local level and include checks against the legal systems in the country of residence.

2.7.2 Staff Onboarding and Offboarding

Digital Asset employs comprehensive onboarding and offboarding processes that include the provisioning and deprovisioning of physical resources, as well as user logon accounts, logical access rights, and data access permissions. User access is revoked on last day of employment. Staff access rights are periodically reviewed for alignment to role or function.

2.7.3 Mandatory Security Training and Awareness

Security is considered a key responsibility of each and every member of staff. The company provides onboarding and ongoing awareness training. Training includes:

- Onboarding security training for all new hires
- Annual Security Awareness training and acknowledgement
- Periodic Phishing testing and awareness
- Weekly Information Security newsletters ("Security Matters")
- Use-case- or SME-related training, as required
- Annual Policy review and acknowledgement

Employees are provided training for emergency situations with CPR, fire drills, and table-top exercises as available.

2.8 Physical Security

Digital Asset requires physical security controls for all office locations. These may be maintained by Digital Asset directly or utilize building management or office services. Each staff member is required to have an individual access token to access the office spaces. Security cameras or CCTV is required on all ingress/egress points and for any internal secure

Technology or IT closet. Ingress lists are produced on a monthly basis and reviewed against current employee lists.

Digital Asset does not maintain its own data centers, but utilizes premier-tier cloud services providers, such as Amazon AWS, Google GCP, and Microsoft Azure. The security of physical access to such services is delegated to the service providers, who can provide the appropriate security certifications and assessments on request.

3 Security Capabilities of Digital Asset Products

Digital Asset works with the world's largest companies to build solutions that synchronize complex multi-party workflows, lower operational cost, and mitigate risk. We combine deep industry expertise with an intuitive Smart Contract modeling language and an extensive partner network to deliver software that harnesses the benefits of Distributed Ledger Technology (DLT).

Digital Asset's Smart Contract language (daml™) and software development tools (daml:Connect) allow clients to focus on solving business challenges and overcoming the barriers to innovation.

The following sections describe the security features of Digital Asset products and solutions built with our partners.

3.1 Daml

Daml is Digital Asset's open source, intuitive smart-contract programming language used to digitize multi-party agreements and automate transactions in a precise and secure manner. Daml enables enterprises from start-ups to large, highly-regulated organizations to achieve more efficient business processes, reduce risk, and develop new products and services that can transform an industry.

Comprehensive, user friendly reference documentation is available that has been circulated, vetted and embraced by all commercial and open source communities. Please read this section alongside our product documentation. In particular, please refer to the Daml Documentation for an overview of Daml, which can be accessed at:

https://docs.daml.com/index.html

The following provides further detail specific to the security features of Daml and distributed ledgers more broadly.

3.2 Distributed Ledger Technology

A Distributed Ledger is a technology that provides data persistence with structure, integrity, and privacy. A Distributed Ledger is one that contains multiple nodes, dispersed across many organizations and locations and, depending on the specific features, offers varying levels of anonymity, privacy, performance, etc.

The specific details of the security capabilities of a ledger are dependent on the implementation chosen. Daml is supported on a growing set of stand-alone and distributed ledger implementations, including VMWare, Corda, Sawtooth, Amazon QLDB, and blockchain databases. The selection of a specific ledger implementation results from the trade-offs of

various functional and non-functional capabilities, including performance, security, resilience, distribution, identity, and trust assumptions, etc.

A Daml Ledger is the combination of the Runtime that executes the multi-party workflows written in Daml and an underlying ledger implementation.

See our documentation (Ledger Model) for the requirements for Daml support.

3.2.1 Daml Ledger

A Daml Ledger is a collection of one or many nodes, which serve different functions:

- Submitter nodes are able to take commands from users and submit transactions to the network, involving interpretation using the Daml Engine.
- Validator nodes receive transactions from the network and validate them using the Daml Engine.
- Reader nodes receive transactions from the network and make ledger data available to users.
- Writer/Committer nodes coordinate the commit protocol and in the process receive and send transactions.

What gets sent over the wire to a specific ledger node or between nodes depends on the individual ledger and its implementation, but once the data hits Digital-Asset-built components, the data is in the protobuf schemas and handled as described above.

Many Ledgers will store the committed transactions from the transaction schema in persistence (Private Contract Store or PCS) and treat that as the essential state from which the Active Contract Set (ACS) can be computed, resulting in a current ledger value or state.

Daml packages are distributed across all nodes that need to execute or verify the transactions. Daml Packages are validated by the Daml Runtime prior to use. Standard change-control processes are executed by node administrators to distribute and activate Daml Packages before transactions involving the package are evaluated.

3.2.2 Common Deployment Pattern

A Daml-based system is a combination of:

- Integration adapters: interface with external systems and users; this might include industry standard messaging, such as ISO20022, CDM, Fix, etc.
- Application(s) (or “Nanobots”): drive the state of the business workflow
- Daml Runtime: executes and enforces state changes based on the rules and transitions defined in Daml. Runtime also interfaces to the underlying ledger implementation for persistent, validation, and distribution.
- A (distributed) ledger: persists and synchronizes that (immutable) state across participants and allows for verification and privacy of data

A typical deployment of a Daml based system is depicted in Figure 2.

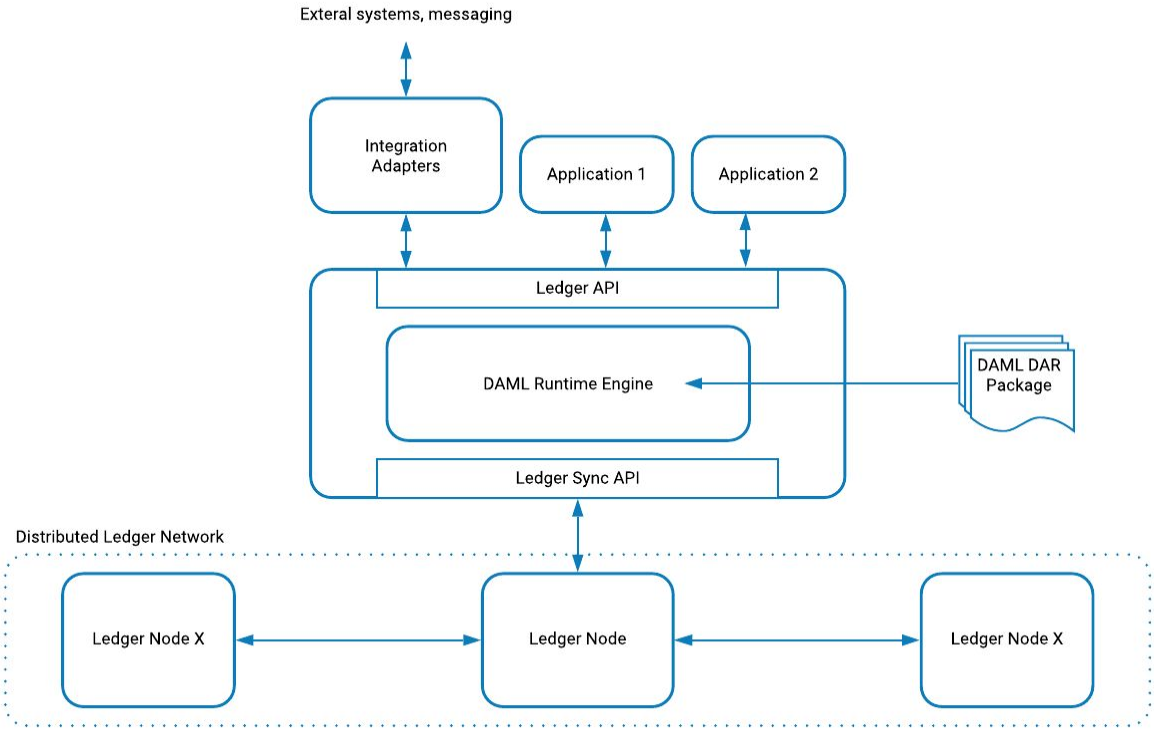

Figure 2: Daml System

3.3 Daml System Security

In common with normal application deployment best practices, it is recommended that a Daml based system be deployed to an appropriately secured environment. Environment security is often unique to each use case environment and is outside the scope of this document.

Depending on requirements and selection of a particular ledger, a deployment might use traditional mechanisms to encrypt data at rest and in transit, including secure connectivity (TLS) or database, file, field, or block-level encryption. System Hardening through industry standard best practices such as those provided by the Center for Internet Security (CIS) must be used.

3.4 Daml Language and Runtime Security

The Daml Smart Contract language is designed for decentralized systems with very low trust assumptions. Trust is the confidence towards participants and the network with respect to how well behaved they are in terms of intention and reliability. The Language, Runtime, and Ledger implementations have accordingly been developed with security as a high priority.

The design of the Daml Runtime provides protection against several broad classes of attacks. The following sections describe attack vectors, risk mitigation, and defense strategies

3.4.1 Daml Language and Packages

DAML is a programming language that is compiled into an intermediate format (DAML-LF) that is interpreted by the Daml Runtime during execution.

The Daml Surface language is written and tested locally on the secured, Dev/Eng-managed endpoint. Committed code is built, tested, and packaged through the CI/CD pipeline prior to distribution or deployment. The Daml Compiler is based on the well-supported open source GHC (https://www.haskell.org/ghc/) compiler[1] with many years of research and community effort behind it. Digital Asset has incorporated extensions to GHC's Parser to add support for language features unique to Daml and components to convert the GHC-internal GHC-core representation to the DAML-LF format.

DAML-LF is an extension of System-F (the polymorphic Lambda Calculus[2]) and is represented in the DAML-LF protobuf schema.

Google Protocol Buffers (Protobuf) is a language-neutral method of serializing structured data i.e.: an extensible way of serializing (something which is being published or broadcast in a number of parts) structured data for use in communications protocols and efficient data storage. It is a popular, supported, widely-used framework.

.DALF Files: A Daml package is a binary-encoded protobuf message of type "Package" as defined by the Google protocol buffers. Daml packages are typically stored as *.dalf files.

The output of the Daml compiler is a *.dar file, which is essentially a Zip archive containing meta data, the *.dalf Daml package of the compiled Daml file(s) and the *.dalf files of any dependencies.

The Daml Packages are uploaded and distributed across ledger nodes at runtime, subject to Change Approval.

3.4.2 Daml Static Analysis (DLint or Daml lint)

Daml Studio is the user-friendly developer environment for Daml Templates and Packages and is provided as part of the Daml Connect. It is based on the open source Microsoft VS Code IDE and provides Daml-specific extensions for workflow development and testing.

[1] **Glasgow Haskell Compiler** is a state-of-the-art, open source, compiler and interactive environment for the functional language Haskell

[2] System F, also known as the (Girard-Reynolds) **polymorphic lambda calculus** or the second-order **lambda calculus**, is a typed **lambda calculus** that differs from the simply typed **lambda calculus** by the introduction of a mechanism of universal quantification over types.

As part of the Daml Studio, Digital Asset provides a consistently enhanced and configurable static analysis tool and linter - DLint - for the Daml Language. The Daml DLint builds on the capabilities of the Haskell equivalent – HLint – a tool that has been in use and with ongoing upgrades for 15+ years.

DLint runs alongside the developer as they code their business workflow logic and scenarios, and it highlights a variety of best practices, including:

- DA Recommended Daml Language Best Practices
- Lambda, Monad, and Recursion handling recommendations
- "Code smells" - *In computer programming, a code smell is any characteristic in the source code of a program that possibly indicates a deeper problem.*
- Reducing code duplication and better structure for templates
- Static detection of potential runtime errors

DLint can be run as a standalone tool as part of the security and quality checks of a Continuous Integration (CI) build pipeline.

Digital Asset continues to enhance this capability as best practices, potential language traps, and recommendations evolve. Digital Asset also works with our partners to provide reference application examples, highlighting greatest efficiencies, best practices, and optimized transaction patterns.

3.4.3 Daml Studio Scenarios

Daml Studio provides developers with mechanisms to define "scenarios" that allow live execution and analysis of Daml workflows as part of the development process. These execute against a local Daml Runtime for validation of business workflows outcomes.

3.4.4 Daml Runtime (Daml Engine)

The Daml Runtime (or Daml Engine) runs Daml code. It is used to interpret commands to turn them into transactions and to validate transactions by "re-interpreting" them.

The current Daml Runtime is an abstract CEK machine written in Scala that runs inside a Java virtual machine (JVM). The CEK machine implements small-step semantics for the lambda calculus by starting with an initial state and taking incremental "steps" to evolve the computation. Like a board game, the CEK machine defines a set of configurations and defines a step function that transitions from one configuration to the next.

The Daml Engine has read access to a Private Contract Store (PCS), which is a persistent key-value store, storing currently active contracts. Keys are Contract IDs, which are restricted, typed strings, while values are binary encoded protobuf messages of type "ContractInstance" from the transaction schema.

The engine is invoked either with an "Update" from the DAML-LF schema or with a transaction from the transaction schema. In either case, the engine will load the referenced Daml packages,

try to fetch any referenced contract from the PCS, and then evaluate DAML-LF to generate or validate a Transaction.

3.4.5 Daml Ledger API

Applications interact with the ledger via the Ledger API, served by the Ledger API Server on Submitter and Reader nodes. The Ledger API Server is a JVM-hosted component. The Ledger API is a gRPC API with messages described by the Ledger API Schema. Applications are authenticated over secure TLS connections.

The Ledger API Server keeps an Active Contract Store (ACS) and a transaction log in persistence. As with the PCS, keys and values are defined in the transaction schema.

What is the advantage of this approach? gRPC allows the definition of services with clear interfaces and structured messages for requests and responses. This model translates directly from programming language concepts like interfaces, functions, methods, and data structures. It also allows gRPC to automatically generate client libraries and is more uniform, more compact, and because of that, more easily secured.

3.5 Daml Runtime Attack Vectors

A well-configured ledger exposes four areas against which a potential attack could be mounted. This section describes how using Daml mitigates risk in those areas:

- Attackers could send malicious messages to the Ledger API, attempting a denial of service (DoS) attack or damage on/to the hosting node.
- Attackers could send malicious messages via the APIs connecting the network nodes, attempting a D0S attack or damage to remote nodes.
- Attackers could upload malicious DAR files, either containing files that are not meta data or *.dalf or containing Daml packages designed to do harm.
- Attackers could attempt to discover and exploit a weakness evident in a particular Daml model.

Outside Daml, components such as language bindings, ledger clients, Navigator, json-api, and similar be targeted. Navigator is a browser-based UI for inspecting and interacting with Daml code running on a local sandbox.

3.5.1 Attacks Against the Ledger API

In production environments, the Ledger API is typically protected with SSL/TLS, behind a reverse proxy that handles authentication for both. Additionally Digital Asset is evolving an authorization solution built directly into the API.

The Ledger API is gRPC based, meaning the messages are decoded from a binary format on the wire to a typed schema by an efficient gRPC infrastructure. gRPC is open source, has wide adoption and community support, and is backed by Google. Message sizes are compact and restricted.

An attack that targets the runtime environment before a valid message from the API schema is handled by Digital Asset's components. These attacks would therefore need to rely on vulnerabilities in gRPC, or the hosting JVM

Special data values like ContractId and Party are checked for special characters, and all interactions with persistence are done through frameworks that additionally protect against injection attacks on the underlying storage (PCS, ACS, or transaction log).

Denial of Service by flooding the API Server with requests is a potential vulnerability, but can easily be mitigated with common protection mechanisms in front of the API, such as rate limiting or throttling gateways or traffic filters.

3.5.2 Attacks Against Foreign Node APIs

Daml is a language that is supported on a variety of ledger implementations, where the communication between network nodes is the concern of the underlying ledger infrastructure. Ledger infrastructure is responsible for communication between nodes and thus for the security of APIs used to communicate between them. The interface between ledger infrastructure and Digital Asset's software components (Daml Engine, API Server, etc.) is via messages from the protobuf schemas described earlier in this document. Thus the same standard protections that mitigate injection attacks on the Ledger API also apply to the APIs used for communication between nodes.

The security of the underlying ledger infrastructure relies on the guarantees by the vendor of that infrastructure. Digital Asset has a set of security claims which cover the underlying ledger and integration. These claims have been audited by independent security auditors and can be made available to ledger providers for comparative analysis.

3.5.3 Attacks via Malicious Daml Packages Files

When uploaded, DAR files are unzipped, all files other than *.dalf are discarded, and the *.dalf files are decoded using standard protobuf infrastructure. All contracts on the ledger reference their associated contract code immutably using a strong cryptographic hash of the defining *.dalf file. Replacing the code associated with a contract is therefore not possible.

Alternatively, an attacker could attempt to distribute a valid, but harmful, DAML-LF package that gets executed in the Daml Engine (e.g., an attempt to cause a DoS via computation in unbounded time and memory). Mitigations for such attempts are discussed below.

Recall that the Daml Engine fetches data from the PCS, which comes with the same protections against injection attacks as the APIs. It then performs a pure computation using an abstract CEK machine to turn the inputs (from the transaction schema) and Daml Packages (from the DAML-LF schema) into a transaction (again from the transaction schema).

Since this pure computation runs in an abstract machine inside a JVM with no low-level data-types like memory arrays, the attack surface is low. The main attack vector is DoS, by

performing computations unbounded in memory or time. This is currently protected by the JVM's resource management. Further work on DoS protection is planned.

A further protection that individual networks or nodes can put in place is to tightly control which packages are accepted and distributed and checking these for potentially unbounded computations. In most enterprise settings, packages are vetted before distribution and production usage.

3.5.4 Issues Introduced by Daml Application Developers

Daml is more rigid than most Smart Contract languages and even most general-purpose languages. It is fully statically typed, all data is immutable, dependencies are always an acyclic graph, all Daml is compiled to DAML-LF, and Daml comes with built-in test tools in form or scenarios.

Many common errors, including whole Mitre Common Weakness Enumeration (CWE) categories (e.g., incorrect casts, buffer over-/underflows, wrap-around errors, indexing/memory allocation errors, errors with pointers, or re-entrancy bugs) are prevented by the Daml language and toolset.

3.5.5 Reporting Issues Discovered in the Daml Runtime.

The Daml Engine is thoroughly tested against a formal specification. Both code and specification are open source under the Apache 2 license and available for independent review and/or formal verification.

Digital Asset has a Responsible Disclosure program in place, under which security vulnerabilities can be disclosed in a secure manner to the Digital Asset security team.

4 Secure SDLC

Digital Asset runs a full CI/CD pipeline for the development, build, packaging, distribution, and deployment of our technologies. The firm has implemented a variety of controls to ensure the security and quality of our products and services. The program continually assesses these controls in light of changes in our product features, services and choice of technology.

4.1 Goals of Secure SDLC

- Security is brought as early in the process as practical.
- All code is tested, reviewed, and approved prior to submission.
- Provenance of all artefacts is understood.
- Licensing of dependencies is understood and approved.
- Vulnerabilities are identified and mitigated during development.
- Opportunities for introduction of Harmful Code accidentally or intentionally are removed.

4.2 High-Level CI/CD Pipeline

Digital Asset follows industry best practices in the setup and management of its CI/CD pipeline (Figure 3). Digital Asset has implemented and continues to evolve a set of controls at various gates of the pipeline.

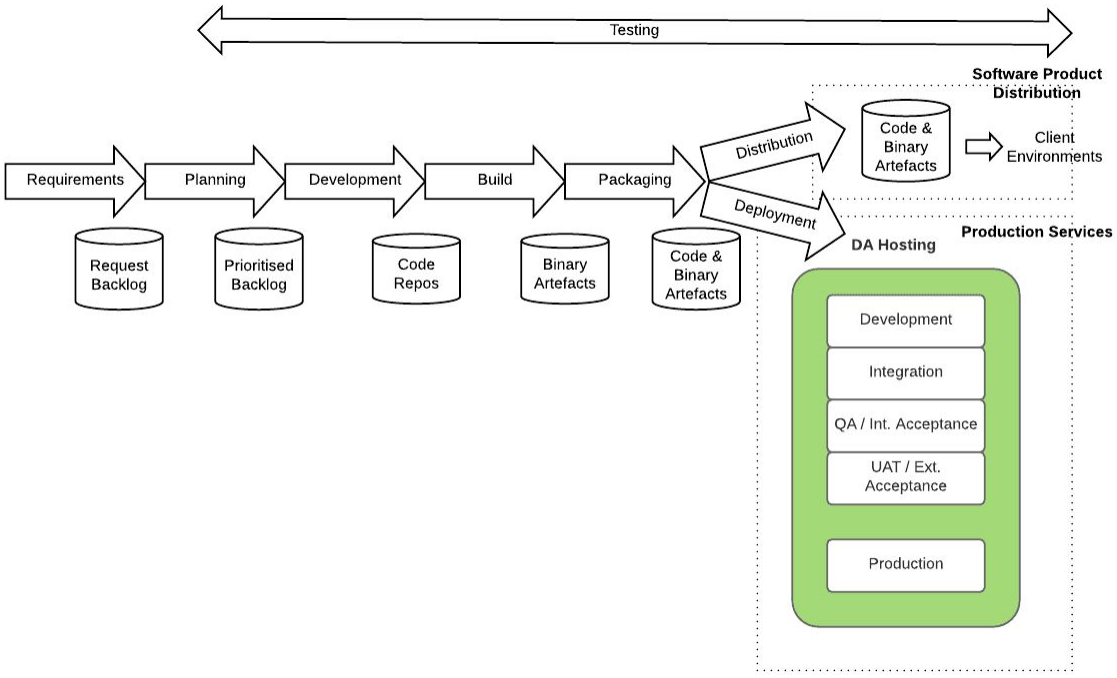

Figure 3: Diagram of SDLC Pipeline

4.3 Source Code Management (SCM)

Digital Asset uses GitHub (private and public repos) as its source-code management system for its proprietary and open source projects. Branch protection rules are enforced for code reviews (see below) and build test-suite completion. For open source projects, CLA (Contributor License Agreements) are enforced for external contributions.

4.4 Build Pipeline Infrastructure

Circle CI and Microsoft Azure DevOps based pipelines are used to build Digital Asset products. Administrative access to these systems (build pipelines, configuration, and infrastructure) is restricted to an approved set of DevOps engineers. Access logs are sent to our SIEM for review.

Sensitive credentials for the build pipeline are managed and secured through the pipeline KMS services and are not stored in source code or configuration files.

4.5 Code Reviews and Testing

All GitHub Pull Requests (PRs) are reviewed prior to being committed to the "master" Branch. GitHub branch protections are used to enforce that all PRs must successfully pass all automated tests (unit, integration, functional, non-functional, etc.) before they can be merged.

In the case of submissions to open source projects, Digital Asset staff are required to review all external contributions prior to merging to the master Branch. No external staff are authorized to approve such code submissions.

Where code submissions may result in production changes (Infrastructure / Application as Code, External Releases), peer review and approval are required as part of Change Management.

Digital Asset continues to improve the efficacy and completeness of its code reviews and testing through new tools and through developer security awareness training.

4.6 Source Code Analysis

Digital Asset utilizes a variety of tools to review its code (CodeDx, Veracode) and also requires peer reviews of all code submitted into the code base. Many Digital Asset staff have backgrounds in security and formal analysis. These reviews cover both functional and non-functional aspects. Digital Asset continues to evaluate new tools in this space to improve the effectiveness and completeness of coverage.

All components and services in a Daml System can be scanned with industry standard tools for a variety of code quality, vulnerabilities, and best practices. In many projects, Digital Asset will

not be developing the components outside of the Daml Runtime, and our partners and clients are recommended to leverage such security tools.

As mentioned previously, Daml code is validated via DLint and scenario execution as part of the development environment.

Digital Asset has utilized third party security design and code auditors to review its products and associated security claims and is planning further engagements.

4.7 Open Source and Supply Chain Risk

Digital Asset has open sourced significant portions of the Daml code base and is also a heavy user of open source components and libraries. Digital Asset uses commercial and open source tools to discover and validate the set of all open source dependencies. These tools allow Digital Asset to identify improper licensing or vulnerabilities in upstream dependencies that are used in Digital Asset products.

Synopsys BlackDuck is used to identify dependencies in Digital Asset products and to scan for software license issues (toxic and copy-left license types) and vulnerable dependencies. All identified issues are ticketed and assigned to relevant product or component owners for assessment and prioritization. Digital Asset can provide a Bill of Materials (BOM) detailing all components in a release and their associated licenses.

Digital Asset also leverages other tools, such as GitHub Repository Vulnerability Analysis and Google Container Registry Analysis to identify dependencies and their associated vulnerabilities in source code and Docker containers.

As this is a rapidly evolving area in the security space, the firm continues to evaluate additional tools and works to understand Best Practices in this area.

4.8 Release Management

Digital Asset software releases are taken from the source repo master branch and built through the automated CI/CD tools. The master branch reflects that latest set of peer-reviewed and tested code.

Code signing is done when pushing to a variety of external artefact repos, and access to the signing keys is controlled within the CI/CD toolset. Only a very small set of engineers have access to the signing keys.

4.9 Containers and Docker Security

Digital Asset uses Docker technology during the development and testing of our products. We use Google Container Registry Analysis to scan for vulnerabilities in published images. We continue to evaluate additional tools to enhance our capabilities vulnerability analysis and runtime access control and protection.

Digital Asset utilizes Kubernetes, Istio, and Envoy, along with Container OS to deploy secure applications in Google Cloud environments. Additional tools are used to monitor for cloud configuration issues.

5. 5 Appendix

Digital Asset partner with a small number of cloud service providers, all annually audited

5.0.1 Amazon Web Services (AWS) Certification

Under constant audit and currently carry the following Certifications/Attestations
1. C5 [Germany]
2. Cyber Essentials Plus [UK]
3. DoD SRG
4. FedRAMP
5. FIPS
6. IRAP [Australia]
7. ISO 9001
8. ISO 27001
9. ISO 27017
10. ISO 27018
11. MLPS Level 3 [China]
12. MTCS [Singapore]
13. PCI DSS Level 1
14. SEC Rule 17-a-4(f)
15. SOC 1
16. SOC 2
17. SOC 3

5.0.2 Google Cloud Platform (GCP) Certification

1 Cloud Computing Compliance Controls Catalog (C5) Information security of cloud services.
2 COPPA (U.S.) Protecting children's online privacy.
3 CSA STAR Securing cloud computing environments.
4 FedRAMP Assessment, authorization, and monitoring.
5 FERPA (U.S.) Protecting the privacy of student education records.
6 FIPS 140-2 Validated FIPS 140-2 Level 1 Certification Implementation for Google Cloud Platform.
7 HIPAA Protecting health information. Click here for the covered products and services under the Google Cloud BAA.
8 HITRUST CSF Industry Agnostic Certification Framework for Regulatory Compliance and Risk Management.
9 Independent Security Evaluators (ISE) Audit Industry Agnostic Certification Framework for Regulatory Compliance and Risk Management.
10 ISO 27001 Managing information risks.
11 ISO 27017 Controlling cloud-based information security.
12 ISO 27018 Protecting personal data.
13 MPAA Protecting intellectual property data.
14 NIST 800-171 Security requirements for United States Federal controlled unclassified information.

15 NIST 800-53 Security and privacy requirements for United States Federal information systems.
16 PCI DSS Protecting customers' card information.
17 Sarbanes-Oxley Act (SOX) Improving the accuracy and reliability of corporate disclosures.
18 SEC Rule 17a-4(f), CFTC Rule 1.31(c)-(d), and FINRA Rule 4511(c) US Record Retention Regulations
19 SOC 1 Controls over financial reporting.
20 SOC 2 Controls over security, availability, and confidentiality.
21 SOC 3 Public report of controls over security, availability, and confidentiality.

5.0.3 Azure Certification

1 CIS Benchmark
2 CSA Cloud Control Matrix
3 CSA-STAR-Attestation
4 CSA-Star-Certification
5 CSA STAR Self-Assessment
6 ISO 20000-1:2011
7 ISO 22301
8 ISO 27001
9 ISO 27017
10 ISO 27018
11 ISO-9001
12 SOC 1
13 SOC 2
14 SOC 3
15 WCAG 2.1