White paper

Digibee Platform Security

Security Protocols to support the latest Integration requirements.

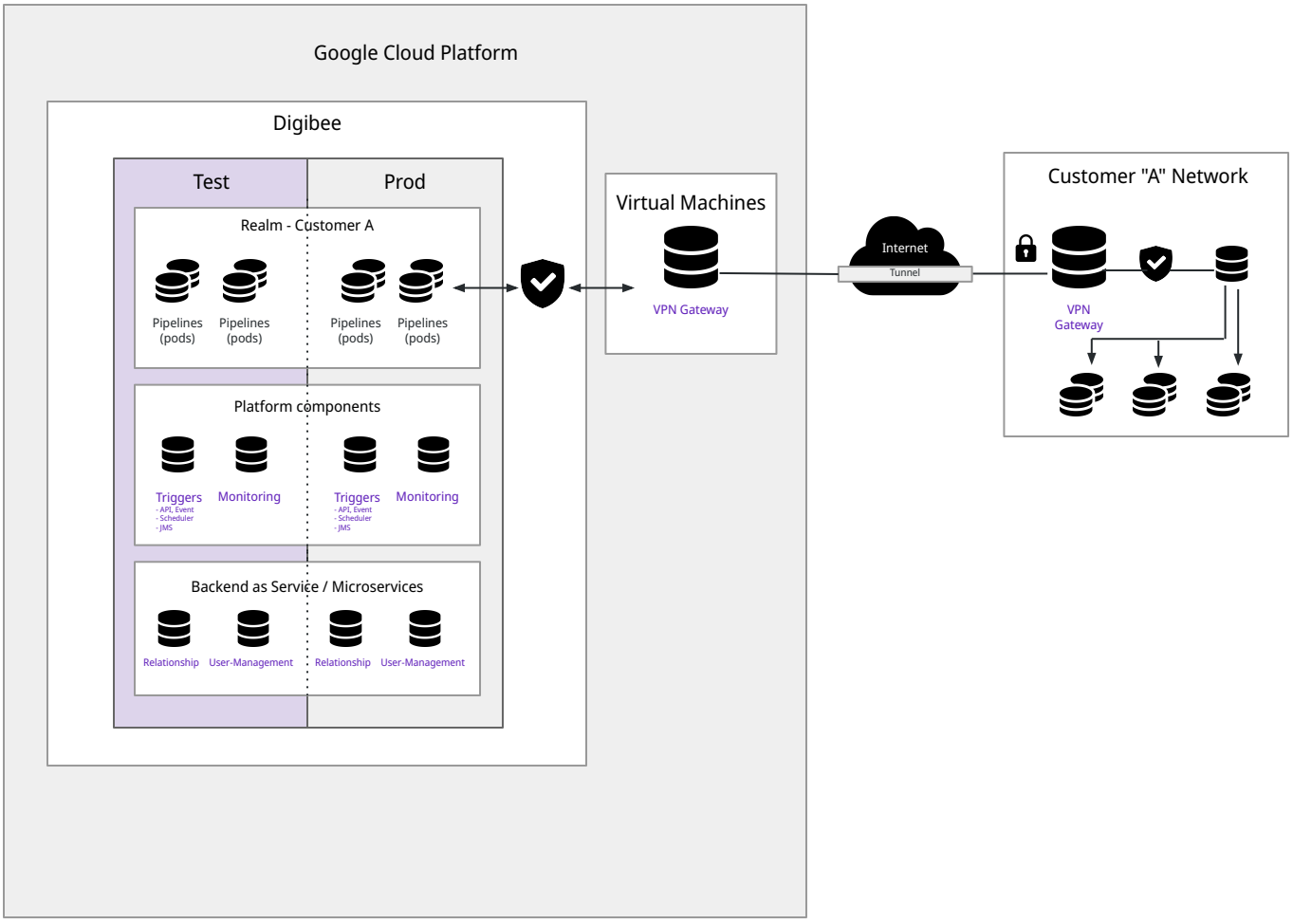
digibee | Enterprise Integration. Redesigned.

For all enterprises, data and system integrity are paramount. The Digibee platform provides a safe and secure environment with security protocols that are well defined, documented, and that meet or exceed industry best practices.

Platform Architecture

The Digibee integration platform is 100% cloud native, available as a SaaS offering or deployed on a private, Digibee-administered cloud. The platform relies on Kubernetes. The pipelines (containers) access the internet to call exposed endpoints such as SOAP and REST.

Digibee employs an isolation model whereby only the Digibee platform application components can call the pipelines. The VPN gateway configuration connects with the customer network, configuring IPTABLES to establish the translation between a specific VPN gateway port and an IP and port within the customer network. All traffic between each dedicated VPN connection is isolated, protected and encrypted (IPSEC).



The platform is hosted on Google Cloud, Amazon Web Services or Microsoft Azure using only tier 1 data centers. All cloud services provide robust controls and procedures based on industry best practices:

AWS Controls | Data & Security | Trust your cloud

User authentication & authorization

Digibee platform can authenticate users by native login (user and password with two-factor authentication) and by integrating the Digibee Platform with a Identity Provider platform using SAMLv2 protocol.

Authorization is implemented using a role-based access control feature which permits the realm Administrator to add, remove or modify permissions for users efficiently. If the realm is integrated with a Identity Provider platform, the realm Administrator can integrate the Digibee's Access Control Groups with Identity Provider platform Groups, to automatically add, remove or modify members of each group within Digibee Platform.

Application authentication & authorization

The Digibee platform administrator may grant, limit, or restrict access to components and pipelines based on authentication and privilege criteria. The model incorporates encrypted passwords using multiple hash passes, access tokens and 0Auth 2.0 as provided by third parties such as Google Services, and other standards.

Application authentication & security

Environment separation is achieved using Test and Production environments. Only authorized users may perform specific actions on any specific environment, subject to the authorization model.

The pipeline lifecycle roles are organized into Design and Runtime phases:

- **Design:** Users create, test, and evolve their pipelines.
- **Runtime:** Pipelines are deployed, executed, and monitored.

Both phases are subject to the authorization model.

Application isolation

Application Isolation employs the following components:

- **Realms:** A security policy domain that enables logical isolation. Each realm defines a separate context where users, accounts, environments, and pipelines are created.
- **Infrastructure / network policy isolation:** The platform relies on Kubernetes, implementing network policies to isolate Pods, rejecting all unauthorized connection requests (incoming and outgoing).
- **CPU and Memory reservation:** Each pipeline runs within a single Pod and each Pod has CPU and memory reserved for it to run, preventing misbehaving pipelines from impacting other pipelines.
- **Exclusive deployment models:** Available as a SaaS offering or as a dedicated cloud for customers that desire an additional level of isolation.

API Security

The Digibee platform provides consistently strong API security, including:

- **Tokens and keys:** A range of authentication modes are supported including basic authentication, JSON Web Tokens, API keys, Oauth2, Google Service Account, AWS Secret, and others. Consumers identify and allow external parties to call the pipeline endpoints, defining the permissions and providing an API key to uniquely identify each caller.
- **Throttling:** Request spikes are managed by configurating throttles to queue requests for future processing.
- **ACLs:** Access to APIs are restricted by ACLs so only authorized consumers can make calls.
- **Payload sizes:** Arbitrarily large payloads requests are prevented from hitting the API.
- **CORS:** Cross-Origin Resource Sharing (CORS) is a HTTP-header based mechanism that tells the browser which origins are allowed to make requests to the server while ensuring that this access is secure.
- **mTLS:**The mutual TLS, or mTLS, is a kind of mutual authentication protocol. It validates that both parties (server and client) have the correct private key, mTLS ensures that the people or systems on both sides of a network connection are who they claim to be.
- **Google Cloud Armor:** A network security service that provides defenses against DDoS and application attacks, and offers a rich set of WAF rules (applicable for Digibee SaaS realms).

Audit

System audits are performed by the Digibee platform to keep track of system activity through system and security logs. User audits track user activity through user activity logs.

Functional Correctness

The Digibee platform has been tested for functional correctness using real-life business scenarios of customers.

Open Source

- **Attribution:** The Digibee platform complies with the fulfillment of notice and attribution requirements for open source products
- **Representations, warranties, and support:** The Digibee platform complies with the representations, warranties, and support requirements for open source products

Encryption

A broad range of encryption measures are in place to protect data:

- **Transport encryption:** Using Transport Layer Security version 1.2 (TLS 1.2).
- **Asymmetric Cryptography:** Also known as public-key cryptography, this system uses pairs of keys: public and private. The public key may be known to others, while the private key may only be known only to the owner. The message can be encrypted using the public key, but decryption is only possible with the receiver's private key.
- **Symmetric Cryptography Connector:** The ability to encrypt and decrypt data through the use of identical keys. (AES-128, AES-192, and AES-256 encryption are supported).
- **PGP Connector:** Support Pretty Good Privacy for encryption/decryption of fields, payloads, and documents.
- **Encrypted credentials and data:** Encryption on persisted data using Advanced Encryption Standard (AES), a symmetric encryption algorithm.
- **Sensitive fields:** Flag fields as sensitive so the data is obfuscated on all generated logs and messages.
- **PBE Cryptography:** Password-based encryption (PBE) is a form of symmetric-key generation that transforms an input string (a password) into a binary encryption key using various data-scrambling techniques
- **RSA Cryptography:** RSA algorithm is an asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes, the Public Key is given to everyone and used to encrypt the data. The Private key is kept private and is used to decrypt the data.
- **Hash Connector:** A hash algorithm is a function that converts a data string into a numeric string output of fixed length. The output string is generally much smaller than the original data.
- **JWT:** JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties.
- **Google IAP Connector:** IAP lets you establish a central authorization layer for applications accessed by HTTPS, so you can use an application-level access control model instead of relying on network-level firewalls. You can define access policies centrally and apply them to all of your applications and resources.