digibee

Resources

Platform Overview

Digibee Hybrid Integration Platform

1

Intro

Digibee presents its hybrid integration platform to drastically reduce the complexity of integration environments and enable companies for digital transformation in practice.

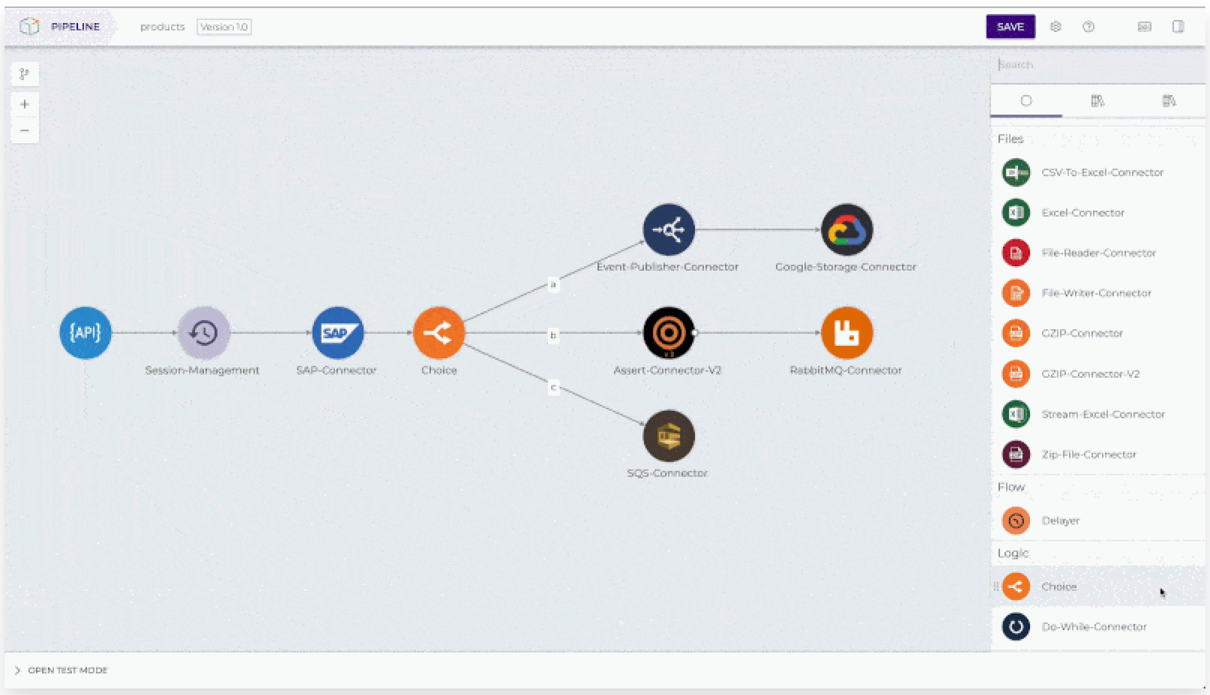
Today's integration teams need access to a myriad of tools and business cases that allow them to mix traditional and modern integration styles. Digibee has the most important capabilities you should evaluate:

- a vast set of components to leverage your integration flows, which might consist of ERPs, CRMs, In-House applications and Legacy systems;
- a resilient, secure and scalable cloud-native architecture built on kubernetes automating your devOps process, version control and simplifying your infrastructure;
- databases and multiple data transformations;
- a set of pre-built business cases to accelerate projects (Digibee Capsules);
- low-code integration approach with a visual, drag-and-drop interface;

- API management support for you to create, secure, manage and share APIs across environments in a quickly and easy way;
- application and data integration driving your ability to copy, transform, transfer and synchronize data across applications;
- messaging and event-driven architectures;
- 24×7 support and all environment managed by HIP specialists;
- always on, digital-first support team;
- specialized professional services on demand.

2

Design



Interface and resources to design the integration

Digibee has a User-friendly and powerful UI that enables the developer to create pipelines* by dragging and dropping components on the canvas and using forms to configure them, what drastically reduces the time invested in the implementation due to the concept of low code.

*Pipeline is the name used by Digibee for each integration flow

Native connectivity and transformation capabilities

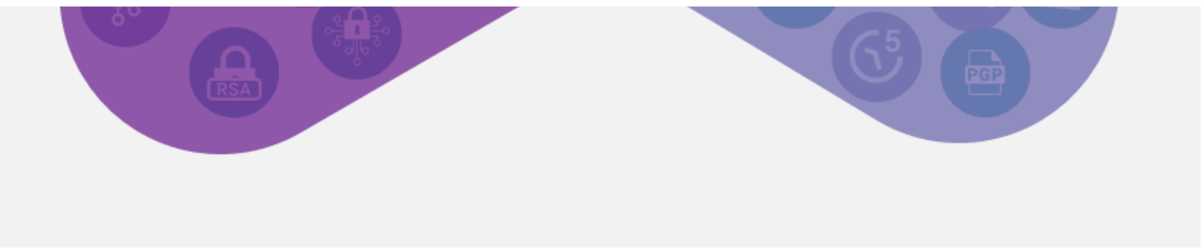
Components are the fundamental building blocks of a pipeline. The Platform provides a wealth of native components. These are our **main components** currently supported:

- Connecting to REST and SOAP Web Services endpoints
- Advanced data transformation
- Data structure conversions
    - XML to JSON
    - JSON to XML
    - JSON String to JSON
    - CSV to JSON
- Data streaming
- Database connectivity (SQL statements and procedures)
- Message broker
    - JMS
    - Kafka
    - SQS
    - RabbitMQ
- File operations
- Cloud-based storage services:
    - Google Storage, AWS S3, Google Drive, One Drive and Dropbox
- ERP integration:
    - SAP
    - Totvs
    - Oracle E-Business Suite
    - Salesforce
    - Bling
- MS Excel integration
- File exchange:
    - FTP and SFTP
    - WebDAV
- Email Sending
- User Management and Authentication
    - JWT
    - Basic Authentication
    - OAuth 2.0
    - Credentials Vault
- Cryptography
    - Symmetric and Asymmetric
    - Digital signatures
    - PGP
    - CMS
- Event Publisher
- Relationships – mapping identifiers between different systems
- Object Store – ability to temporary store data in the Platform for multiple integration use-cases
- Assertions
- Custom Scripts (Javascript)
- Robotic Process Automation scripts
- Structure and data message validators
- Logic
    - Loop
    - Iteration
    - Conditional processing
    - Retries
    - Parallelism

Capsules

Capsules are reusable components that any user in the Platform can create by applying the same visual development model conceived in the pipeline creation. It allows the definition of integration flows that will be published in the components palette for later use.With capsules, organizations can offer pre-packaged business logic that can be used by internal teams, customers and partners.

With capsules, organizations can offer pre-packaged business logic that can be used by internal teams, customers and partners.



Relationship Management

Whenever a business process is implemented, keeping data consistency across systems is a challenge – for instance, a product is represented in different ways by an e-commerce system and a warehouse management system (different IDs, different attribute names); still, they represent the same physical product that needs to be delivered to a customer. Relationship Management allows creating mappings between different systems, providing both data and process consistency while greatly simplifying pipeline creation.

Object Store

When building integrations that touch multiple systems, it is not uncommon to have to rely on data staging areas, such as temporary tables. The object store lets you insert, update, search and delete JSON documents within collections, providing this much needed functionality in a structured, efficient and easy-to-use manner. A common usage pattern for the Object Store is implementing a transaction queue: transactions are saved as documents so that they can be sequentially processed as possible or needed.

Multi instance pipelines

Replicating a process across a large chain of stores or branches (here we mean actual stores or organization branches) is a significant integration challenge. By doing that, consistency and a low cost of ownership is even harder to be kept. Multi instance pipelines were created to solve this problem. They consist of a single pipeline with a parameter map that holds all specific information for stores or branches that need to be integrated. Any process changes are made only to this single pipeline, but deployments can be made separately for each store, providing resilience and making it easier to expand the customer operation in a managed, monitored way.

Flexible triggers

Every pipeline needs a trigger for its execution. There are many different trigger types in the Platform:

- expose the pipeline to direct API call (Rest or HTTP endpoint);
- schedule it for recurring execution (Scheduler);
- associate the pipeline to a registered event, allowing for asynchronous execution – more details on the Events session;
- configure the pipeline to listen to a JMS topic or queue (ActiveMQ and OracleAQ);
- listen to messages from a RabbitMQ broker;
- consume messages from a Kafka topic;
- listen to emails in a IMAP mailbox.

Events

The Platform was designed according to the event-based paradigm. It means that pipelines generate and consume events, creating a fully asynchronous and resilient environment. Events can be managed, correlated and re-executed according to specific customer business needs.

Safe, agile testing

Every pipeline can be executed in test mode, calling real-world endpoints and systems and providing execution logs and messages from within the pipeline design canvas. It enables fast validation, making it easier to make adjustments and corrections without the need of redeploying the pipeline to an execution environment.

Robust versioning

Native, non-optional versioning – it generates minor versions for small changes, major versions for changes that impact the pipeline inputs and outputs. This strategy preserves the version integrity and enables future pipeline evolution.

Sensitive information masking

Data that should not be exposed can be tagged as sensitive data, so it is obfuscated in every platform output (logs and messages).

Auditing

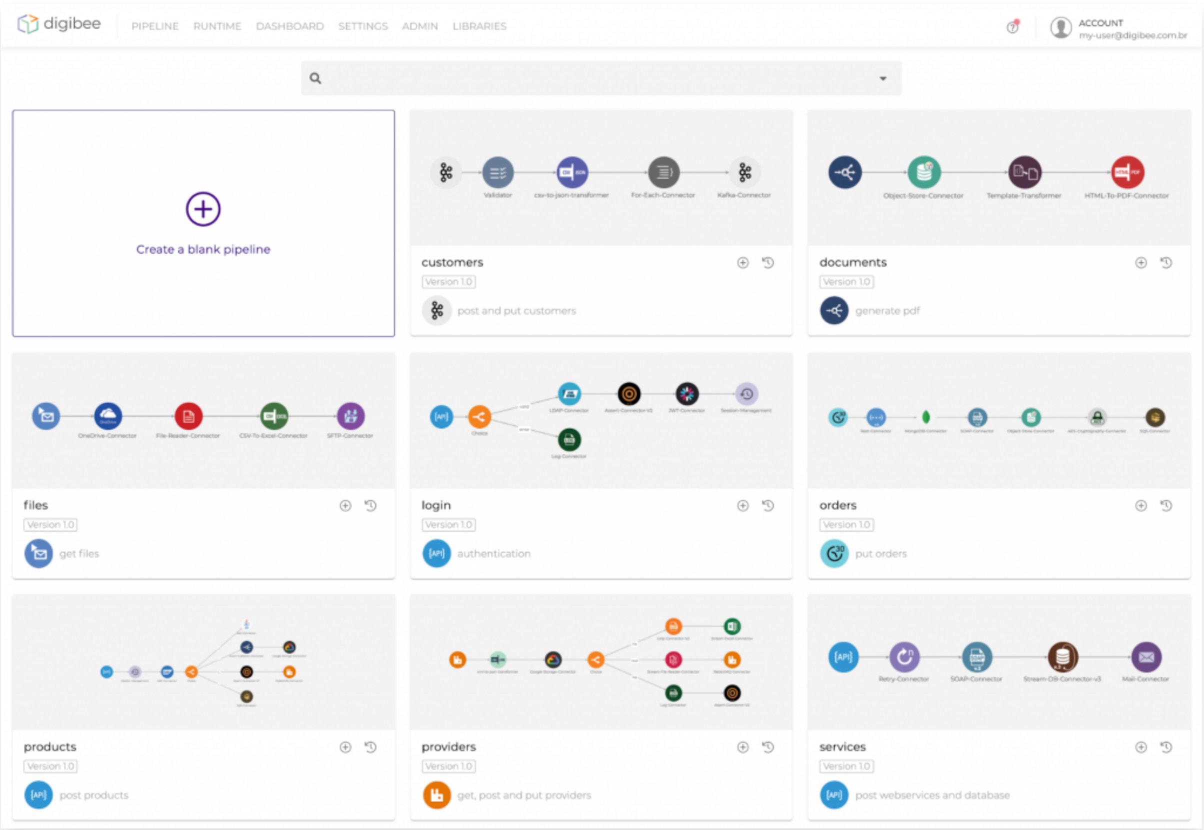
Every operation is audited and stored securely by the Platform, so it can not be inappropriately changed.

3

Deploy

Automated deployment

Deployment is made in seconds: choose the environment – Test or Production -, select the pipeline version and the deployment size. The Platform will create a corresponding number of pipeline replicas according to the selected deployment size, enable monitoring and publish the pipeline, making it immediately available.



customers
Version 1.0
post and put customers

documents
Version 1.0
generate pdf

files
Version 1.0
get files

login
Version 1.0
authentication

orders
Version 1.0
put orders

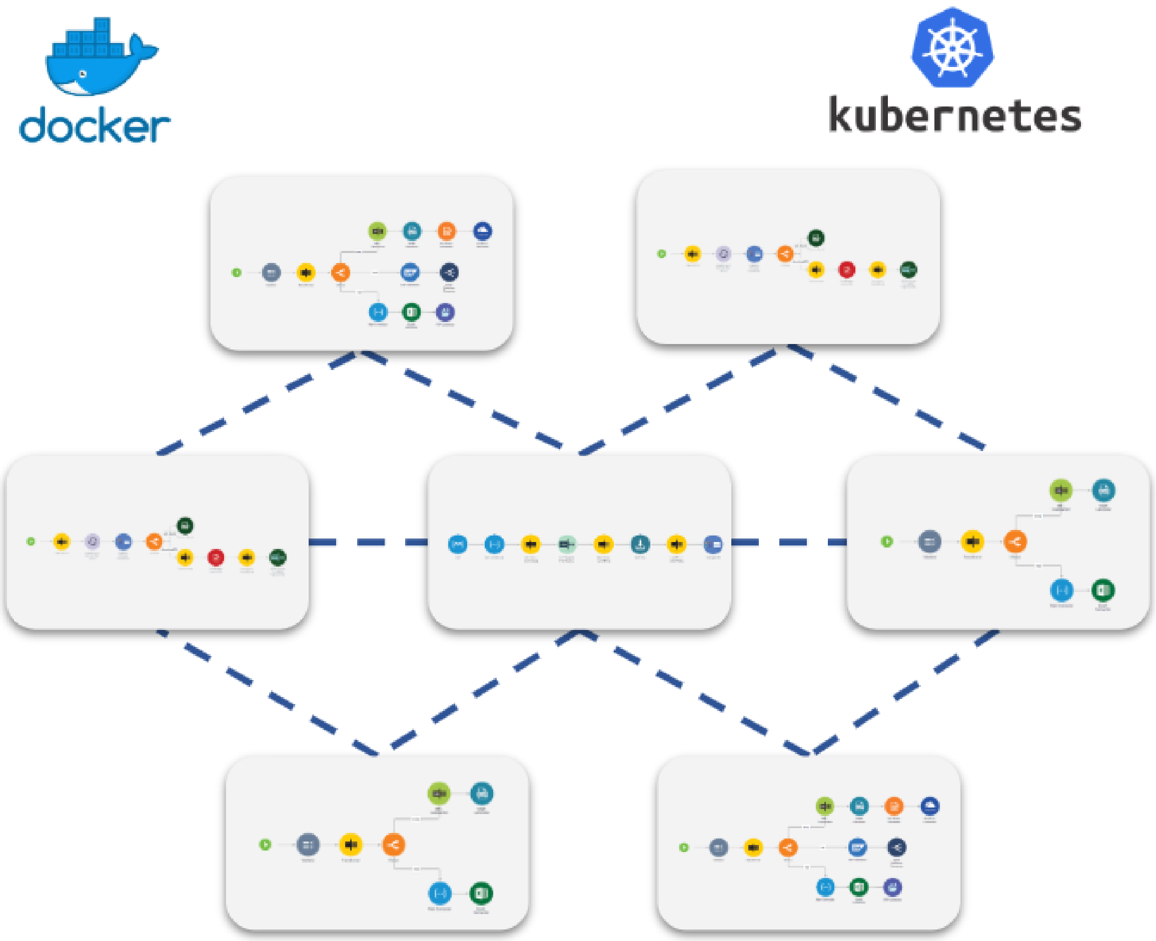
products
Version 1.0
post products

providers
Version 1.0
get, post and put providers

services
Version 1.0
post webservices and database

Segregated Environments

A pipeline can be deployed on non-production environments, making it easier to validate it in runtime. When a pipeline is ready, all you have to do is to deploy it in the production environment.



Resilience

The Platform is 100% cloud native, fully based on containers orchestrated through Kubernetes. A pipeline is deployed through replicas, which are identical execution instances, yet logically isolated into Pods. When an event or a request associated with a pipeline is detected, an available replica will pick it up for immediate processing.

In the event of a replica experiencing some critical error that prevents it from successfully finishing the request processing, the request will be handled by another available replica and processed. The misbehaving replica will be automatically recycled so it is back online, waiting for new requests.

Performance

The Platform architecture promotes process isolation. This means that a running replica does not impact performance or stability of any other replica associated, or not, with the same pipeline.

Scalability

The Platform architecture allows adjusting the number of replicas for the pipeline to be implemented, given the request specific processing requirements. Each replica in our SaaS environment runs in separate zones.

4

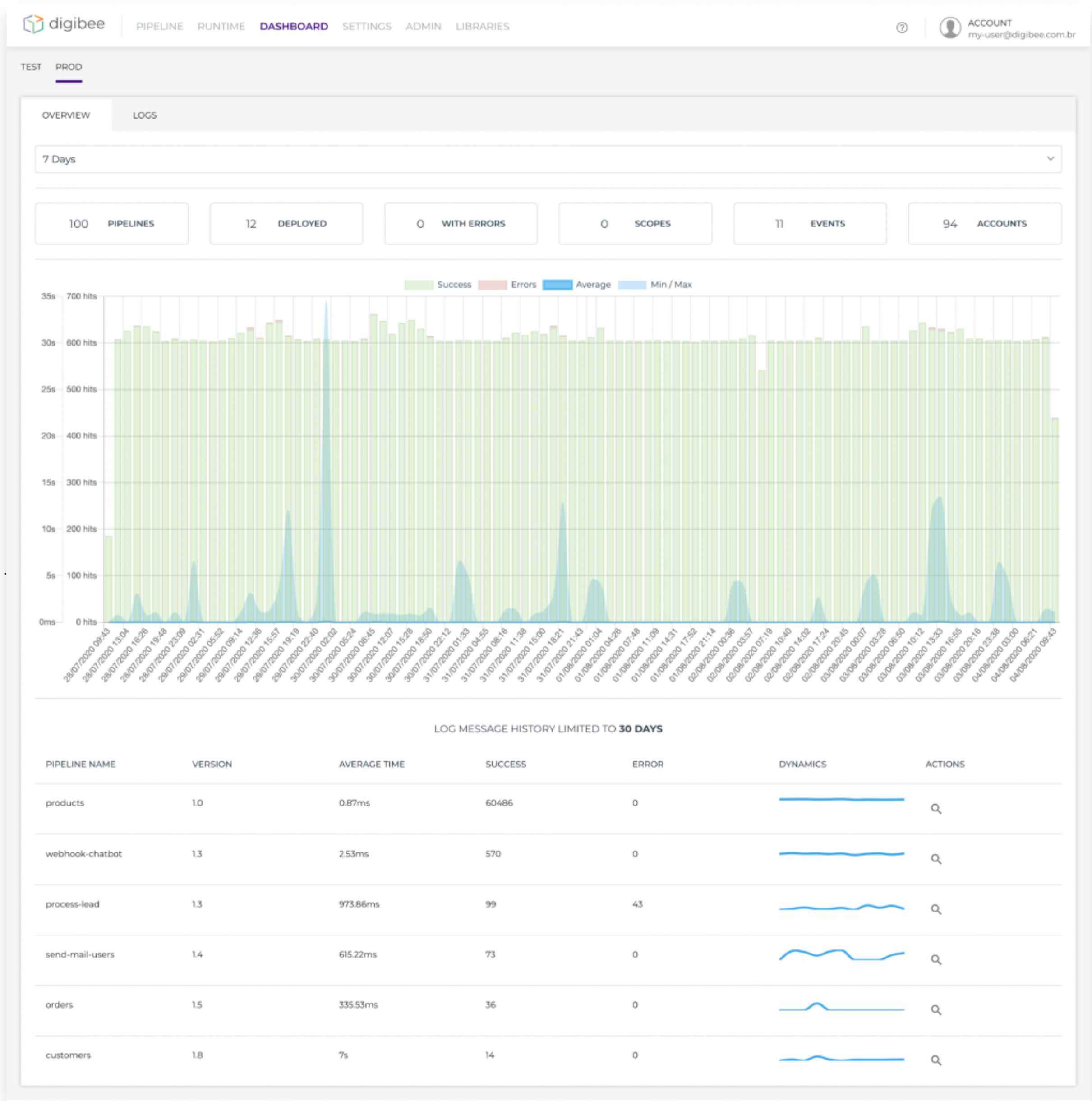
Monitor

Native Monitoring

From the moment a pipeline is deployed, monitoring is automatically activated – no human intervention is needed.

Dashboard

The Platform dashboard provides graphical representation of the pipeline's execution behavior: deployed version, average execution time, errors and execution dynamics as well as access to execution logs.



100 PIPELINES | 12 DEPLOYED | 0 WITH ERRORS | 0 SCOPES | 11 EVENTS | 94 ACCOUNTS

LOG MESSAGE HISTORY LIMITED TO **30 DAYS**

PIPELINE NAME	VERSION	AVERAGE TIME	SUCCESS	ERROR	DYNAMICS	ACTIONS
products	1.0	0.87ms	60486	0		🔍
webhook-chatbot	1.3	2.53ms	570	0		🔍
process-lead	1.3	973.86ms	99	43		🔍
send-mail-users	1.4	615.22ms	73	0		🔍
orders	1.5	335.53ms	36	0		🔍
customers	1.8	7s	14	0		🔍

Logs

Each execution generates detailed logs providing execution time, request and response pipeline messages. Additional logs can be created by the pipelines to support specific business requirements.

Alarms

The Platform generates events that represent pipeline specific conditions. These events can be consumed and sent to third-party ticket management and monitoring solutions.

5

Evolve

Versioning and history

From the pipeline version history it is possible to generate a new version of the pipeline, which can be tested and evolved in the Test environment while the production version continues to run, unaffected.

Coexistence and evolution

The Platform architecture allows different major versions of a pipeline to be running simultaneously in production, enabling adoption of coexistence and zero-downtime strategies.

Ready for Agile teams

The Platform unleashes the benefits of Agile development. I would change the extract to:
“Its intuitive UI turns rapid prototyping and componentization into an agile reality so that teams can move forward without being impacted by interdependencies that, most of the time, prevent them from delivering on their commitments.

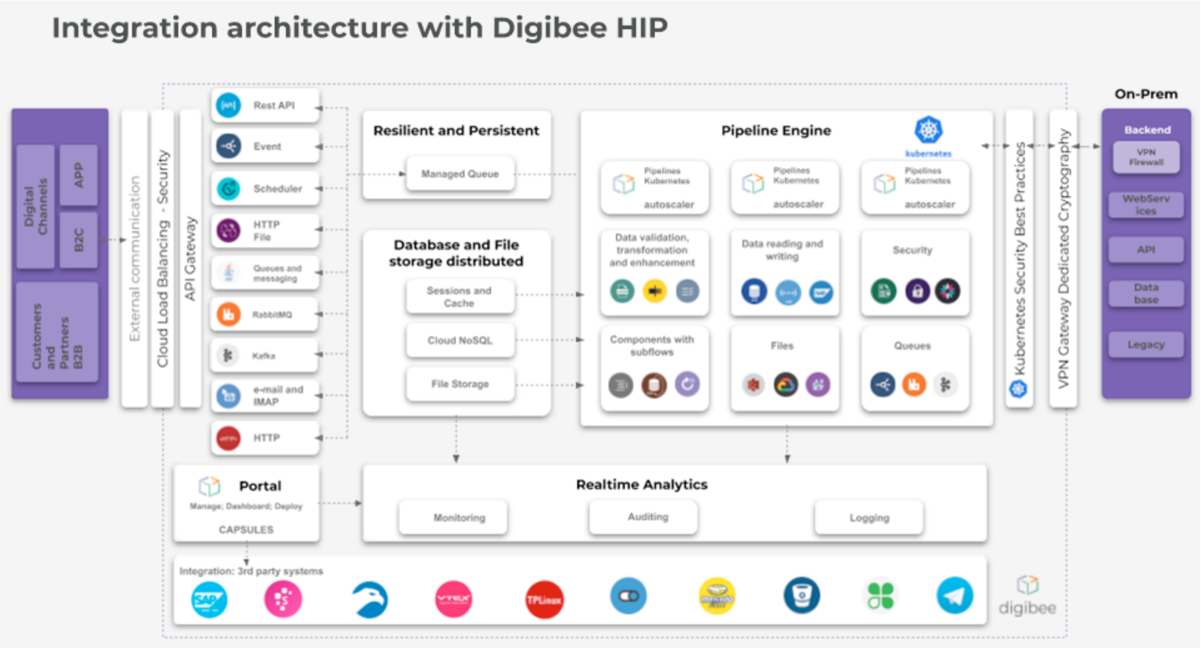
Differentiated support model

Moreover, the Digibee Hybrid Integration Platform creates an environment that enables building integrations collaboratively. Our support model creates opportunities for agile teams to interact with our integration consultants in real time, either to get the answers they need, discuss best practices or to get support to develop and implement their critical integration processes.

Besides, the Platform availability information is publicly available and customers can subscribe to our status page so they can be notified in case of any Platform component unavailability.

6

Architecture



Cloud-Native

Digibee HIP is a 100% cloud-native Platform. It runs on Kubernetes, a proven execution platform that provides huge resiliency and scalability.

The Pipeline Engine

It is the the core element of the platform - think about it as an Integration Runtime. The pipeline engine is responsible for executing all the deployed integrations. It was designed and extensively tested to deliver performance, resiliency and reliably execute your business processes.

Let's understand a little bit more about its inner workings.

Every integration is interpreted by the Pipeline Engine and executed in isolated containers. That means each pipeline has a dedicated CPU and memory capacity allocated to its execution. That prevents a misbehaving integration from impacting other integrations' performance and stability.

Thanks to Kubernetes, when a problematic execution is detected, the offending integration is automatically restarted to be up and running again in a matter of milliseconds. To ensure high fault-tolerance, every integration pipeline is executed on multiple availability zones.

This is very different from traditional integration solutions such as ESBs, where all integrations share the same execution context.

The Pipeline Engine has all the necessary code to execute any component available in our Platform. There is no need to write additional code to connect to a supported technology.

We offer components for message processing, flow control, web protocols support such as SOAP e REST, file manipulation, data manipulation for both relational and noSql databases, security, cryptography and many others.

The Triggers

The Pipeline Engine itself can't communicate with the external world. It needs a trigger to invoke it. The Platform offers a great variety of triggers such as API/REST, Event-based, Scheduled, Message Queues, Email and HTTP.

For both components and triggers, we have a dedicated development capacity for creating new versions to continuously expand the Platform capabilities and provide increasingly support for more enterprise integration scenarios.

To better illustrate the trigger concept, let's walk through some use scenarios:

API/REST

A company wants to offer its developers specific services through API calls. In that case, the integration pipeline needs be configured with a REST trigger so, when it is deployed, an endpoint will be exposed so users can call it. Requests submitted to this endpoint will then be forwarded to the pipeline for processing.

Scheduled Execution

An integration reads files on an SFTP folder, processes them and inputs its data into a system for statutory reports generation - and it needs to run once a day. This can be accomplished by configuring it with a Schedule trigger so its execution is programmed to run every day at 1:00 AM.

Message Queues

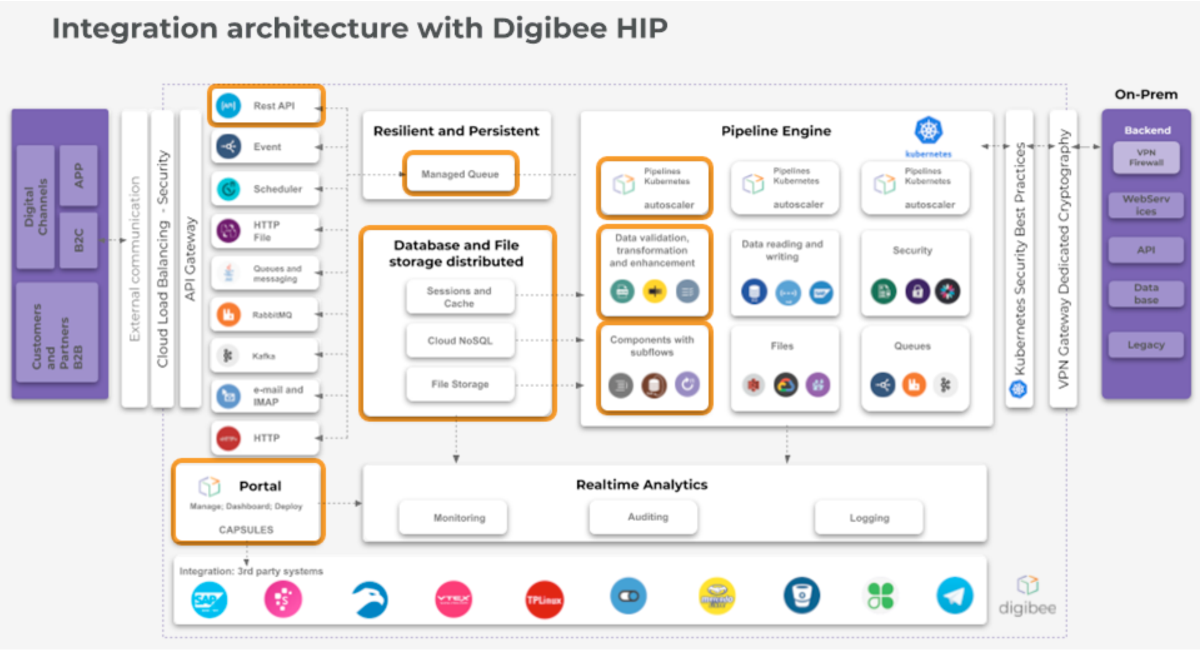
An integration needs to process all messages posted to a specific queue or topic of a message broker software (think RabbitMQ, Kafka). All it takes is configuring the integration pipeline with a Message Queue trigger so it will process messages as they arrive.

The Managed Queue

We have implemented a native queueing mechanism. Whenever a trigger receives a request or is scheduled for execution, it puts a request on the corresponding pipeline message queue and that pipeline is activated as soon as the message arrives. If the integration fails to process the message and is automatically restarted, the message is not lost - when the pipeline is back online, it will catch up and process the pending messages. This provides a high level of resilience, making the Platform virtually immune to execution failures.

Putting it all together

Now we are going to follow a transaction from the moment the customer submits it until the moment the Platform provides a response. Let's assume that transaction is an API call.



The Platform offers a built-in API gateway, protected by cloud provider security features to avoid any attacks such as DDoS. As soon as the transaction goes through the provider edge infrastructure it is received by our gateway that routes the message to the corresponding trigger - in this case, the REST trigger.

The message is posted on the pipeline queue. The pipeline receives it and process it according the designed processing flow - it can manipulate the message, make decisions based on its contents, transform it and enrich it with data obtained from other sources. For enhanced performance and functionality, the Platform also has dedicated advanced caching services and a temporary object storing system, which is called Object Store.

After the message is processed, the pipeline provides the response to the submitted request.

Designing, Deploying and Operating Integrations from the User Perspective

All this process can be monitored through the Platform portal. The portal enables users to:

- build pipelines by using the pipeline canvas, that lets you drag and drop components and draw the integration flow;
- test pipelines through the integrated test-mode functionality;
- deploy pipelines to Test and Production environments;
- monitor transactions and message contents;
- access the audit logs.

Credentials Governance

Every credential necessary to access the systems that needs to be integrated are stored in the Platform credentials vault. Once a credential is created it cannot be directly read by anyone - it can only be accessed by the pipeline during runtime. This strategy prevents direct access to credentials, providing exceptional security and governance: customers can create credentials and share them to pipeline developers for building purposes only - the credentials content will not be available for visualization nor edition.

Connectivity to customer environments

To be able to access resources that are within internal customer networks we offer dedicated VPN gateways that are completely isolated through network policies.

7

Security

Platform features

- The Platform has a series of security controls:
- audit for all administrative actions;
- 2FA for platform access;
- complete user lifecycle management with responsibility segregation;
- ability to integrate user management with customer’s own Active Directory or others;
- best practices for endpoint exposure by using IPS and WAF mechanisms of the major cloud providers;
- 24×7 monitoring;
- real time incident reporting via statuspage.io;
- CPU and memory reservation for each pipeline;
- infrastructure isolation for each pipeline.

Pipeline features

- All the pipelines can be enhanced with security to accommodate the business and technical needs. Here are some possibilities: authentication following market standards;
- cryptography;
- sensitive fields;
- password Management;
- rate limiting;
- throttling;
- IP Restriction;
- ACLs;
- payload sizes;
- redundancy;
- recovery.

To know more about Digibee Hybrid Integration Platform, click here and access our documentation.

Office locations

US

1398 SW 160th Ave, suite 106
Sunrise - FL - 33326

WeWork - The Hub
3601 Walnut Street
Denver - CO - 80205

Brazil

Alameda Vicente
Pinzon, 54,
9ª andar, Vila Olímpia
São Paulo - SP

Follow us:

Talk to us:

Contact
Finance
Support

Digibee platform status

Available on the SAP App Center

2020 - Digibee ®
All rights reserved
Privacy Policy