

PITCHLY PAAS DATABASE AND SAAS APPLICATION LIBRARY

TECHNOLOGY OVERVIEW

PRODUCED BY: PRODUCT@PITCHLY.COM

INTRODUCTION

Our application is a multi-tenant software-as-a-service platform for professional service providers to store information about client experiences. This information takes many mediums including digital characters, digital images, and digital files. This document provides a technical overview of our platform. If you have any questions or would just like to say hi, please contact us at product@pitchly.com.

MULTI-TENANT CLOUD ARCHITECTURE

MULTI-TENANT

Our product is a single instance of software that allows customers (or tenants) the ability to customize the design and information requirements of their database and applications without affecting other tenants. This configuration is known as software multitenancy.

CLOUD ARCHITECTURE

Our application is accessible through any laptop or desktop computer released in the last 10 years. Our platform is hosted on cloud server infrastructure provided by Amazon Web Services. This infrastructure processes and stores data that is made accessible to users through a web browser over the internet. The access of a hosted software product through a thin client (computer) via a web browser is also known as a Software-as-a-Service delivery model.

MICROSERVICES ARCHITECTURE

Applications built on our experience platform are self-contained microservices that can be installed, managed, and deleted independently from the core experience platform. Applications from our library are installed onto a specific database and integrate functionality into the core platform and between integrated databases. Enterprise account holders are provided with Developer access to use our reactive API library to build custom applications that connect to experience records, user permissions, and leverage our robust search and filter functionality.

CUSTOMIZATIONS

SINGLE INSTANCE

Our application is a single instance code base, meaning we maintain, enhance and update only one product which is transmitted to many customers through the internet. This one (product) to many (customer) delivery model improves the scalability and reliability of our service and provides customers the peace of mind knowing they are using the same infrastructure and product as other customers.

CUSTOMER CONFIGURATION

Within the application customers can configure the product to meet their needs. Customer administrators can invite additional users, add/remove information fields, and determine other settings beneficial to their use of our application easily.

NEW FUNCTIONALITY

We are continually improving the product and have an ambitious roadmap of future product enhancements and features. Before we begin development of a new feature we request feedback from the customer who suggested the feature and customers who we feel will benefit from its release. This process ensures every new feature we are building is valuable to each of our customers and not just a select few.

PAY FOR PRIORITY.

We allow customers to prioritize the development of a particular feature ahead of other planned features through a pay-for-priority model.

MAINTENANCE AND UPDATES

MAINTENANCE AND UPDATES

We regularly perform maintenance on our product to remove bugs and ensure our product is up to date with the latest technologies. Most of this maintenance occurs in real-time and does not result in the application being inaccessible to customers. In situations that require a substantial maintenance update the application may be unavailable for a short period of time. During 2016 all maintenance updates were performed without downtime, but should the need arise customers will be notified in advance and scheduled maintenance will be performed outside of business hours.

QUALITY ASSURANCE

We employ a thorough quality assurance program that involves both test automation and unit tests of proposed changes to the product. This automated and iterative process includes a sandbox version of the application so our engineers can thoroughly test new code before it is deployed.

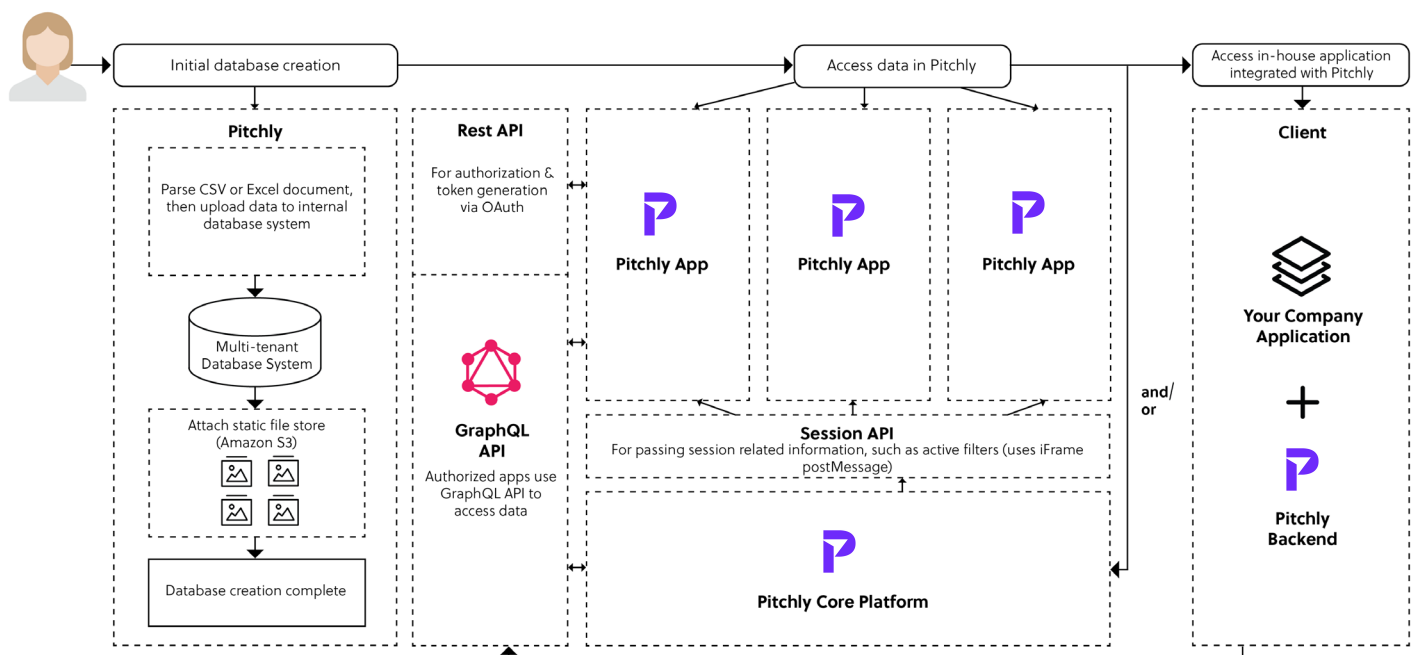
DATA FLOWCHART

Pitchly is built with cutting edge real-time technology. Today, we offer two ways to connect customers' business data to Pitchly.

PUBLIC CLOUD DELIVERY

A description of this process is provided below from left to right:

1. A Pitchly base (think excel worksheet) is created by uploading data. This data is saved on Pitchly's servers.
2. Users can interact with the resulting base and connected apps. These interactions trigger GraphQL API queries to the data on the platform.
3. Conversely, customers can create custom applications using these same API endpoints to develop their own solutions. Customers can now take advantage of Pitchly's single source of truth while retaining the ability to grow.



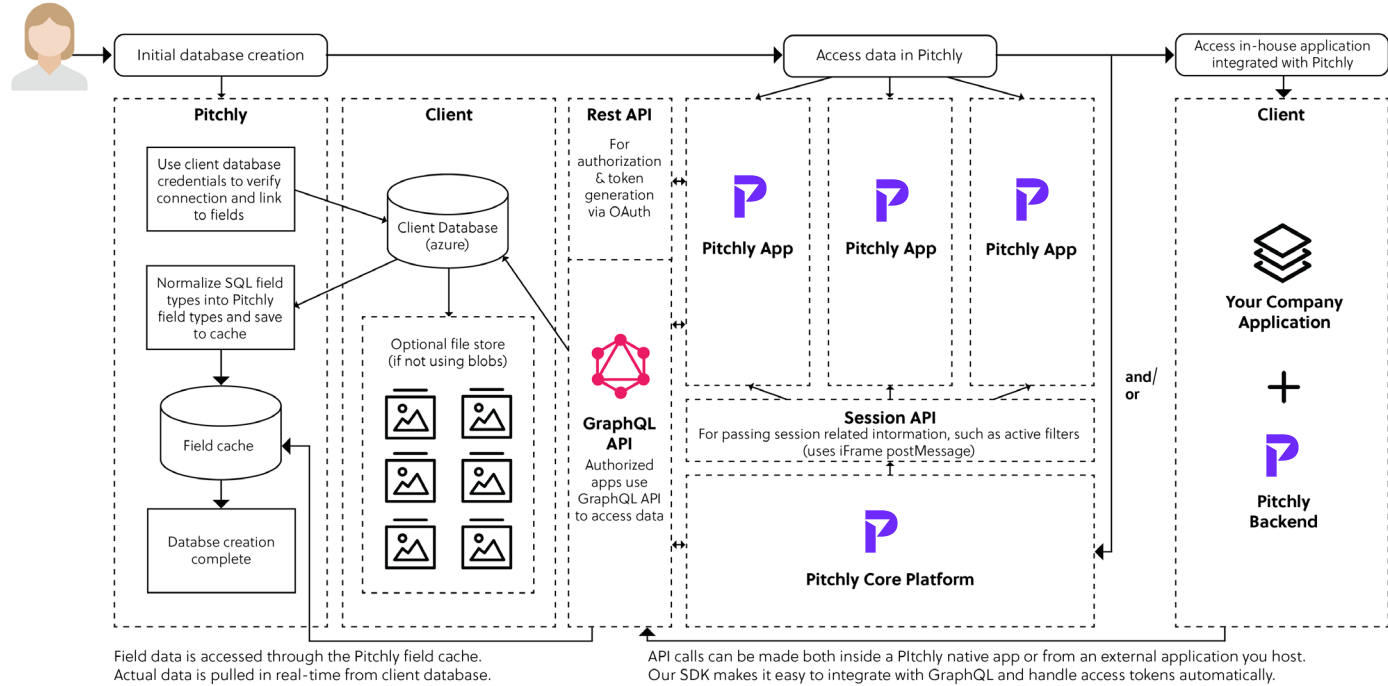
This process involves engaging one of our Professional Service team members to scope the feature being requested to develop a fixed-cost statement of work. The customer then agrees to the statement of work, pays the amount due and has "paid for the priority" of developing a feature.

CUSTOMER FEEDBACK

We love talking to our customers regularly and encourage our users to submit feature requests or report bugs they have found when using our application. Features requests and bug reports can be sent to product@pitchly.com or submitted through our live chat feature within our application. Each ticket is recorded and prioritized according to other user feedback. The user submitting the ticket will be notified when the reported bug is fixed or if we are considering development of the feature requested.

HYBRID ON PREMISE DELIVERY

The second way customers can connect data to Pitchly is through a private or “hybrid” cloud model, as shown below.



In the hybrid deployment a client-side data source (on-premise behind customer firewall) is connected to transmit data to the Pitchly application. Pitchly now acts as an interface to the customer’s data located on their own servers. This includes all connections to Pitchly applications, since data from the customer’s data source is normalized across the entire Pitchly platform. Data is not retained or stored by Pitchly and remains client-side.

FLOWCHART DESCRIPTION.

The flowcharts displayed above depict some of the components of our application. Our application architecture is composed of several key components:

- **Computational.** This includes data processing functions for data ingest and to optionally connect to outside databases or services hosted or controlled by the client.
- **Data.** In our public cloud model, data is hosted by Pitchly. In our private cloud model, data is hosted by the client. In our hybrid cloud model, databases in Pitchly can be mixed between being hosted by Pitchly or being hosted by the client.
- **Storage.** Static assets, which include images, documents, logos, or general files may be stored in Pitchly, stored by the client, or passed through (similar to data retrieval in our private cloud model) to retrieve assets from the client through the Pitchly application.
- **Application.** The Pitchly application itself, which encompasses primarily the platform but also the architecture necessary to deliver microservices provided by Pitchly Apps.
- **API.** Our API controls access to resources via the platform, which may consist of data hosted by Pitchly or hosted by the client. The flow of data to specific applications is controlled by the administrator(s) of the client account in Pitchly, and data is transmitted in a unified format, regardless of the source. Our OAuth API handles the handshake authorization to allow access to certain resources by an application. Our GraphQL API provides access to those resources, given a valid access token provided during the OAuth handshake method. And finally, our Session API simply provides Apps a way to access session-specific information related to the user, such as the current filter or row selections. This information is also restricted to whitelisted origins for a given application.
- **Microservices.** These can consist of either Pitchly Apps that live primarily inside the Pitchly platform, alongside a database, or a client-built application that takes advantage of Pitchly's APIs to deliver custom functionality using data or services provided by Pitchly. Microservices may also be a combination of both.

These components provide a highly scalable architecture accommodating a high level of configuration within our platform and applications, while providing our customers the opportunity to build custom applications to expand the use case of Pitchly for their organization.

SECURITY

We understand your enterprise data is private and we have serious security to make sure it is treated as such. The image below depicts how information exchanged in our product is secured through each interaction in our application. Each icon is explained in further detail following the image.



User accounts are secured through administrative controls to manage users, access and activity. Strong user authentication with password, email and third-party verification ensures only approved users can access the application.



All transmitted data between clients and Pitchly is encrypted with AES 256 SSL encryption, and if client data sources are utilized, encryption is required and IP address whitelisting can be used at request.



Passwords are hashed using an industry standard hashing algorithm with salts and key strengthening. All information stored in Pitchly's databases and file servers is both encrypted in transit and at rest using AES 256 encryption.



Data is stored within ISO 27001 certified data centers, the same standard employed by banking institutions.



Amazon Web Services infrastructure provides industry-leading secure scaling of our application, database and file servers.

Thank you

We look forward to talking to you soon. If you would like more information about our service, please don't hesitate to contact us.

[CONTACT PRODUCT TEAM](#)