



Architecture Document

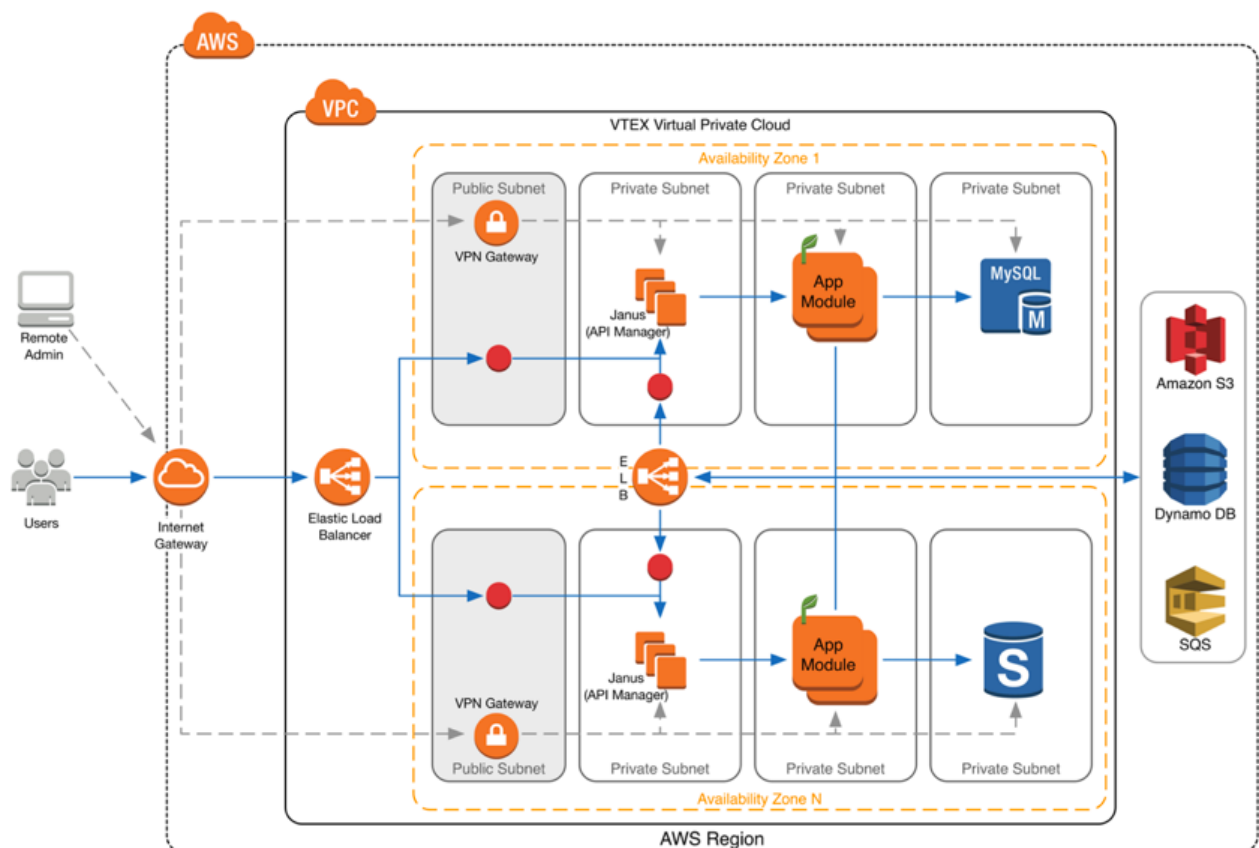
Table of Contents

Overview	3
The Request Journey	3
ELBs	4
Janus	4
App Module	4
Data Services	4
Security	5
Application Security	5
Authenticating Users in VTEX: VtexID	5
Authenticating Applications	5
Authorization	6
Data Encryption: Key Strength & Algorithms	6
Infrastructure Security	6
Firewalls Policies & Rules	6
Vulnerability Patch Management	7
Server OS Hardening	7
Information Security Policy	8
Development & Deployment	9
Team Governance	9
Continuous Deployment	9
Monitoring & Testing	11
Unit Testing	11

Integrated Tests	11
Acceptance Tests	11
Regression Tests	12
Load Tests	12
Logging	12
The Health Check Robot	13
Disaster Recovery	15
Application	15
Data	15
Process of Information Deletion	15
Integration	16
Regulatory Compliance	17
PCI DSS	17
SSAE 16	18
IFRS	18

Overview

VTEX Commerce Suite is a multitenant SaaS solution, based on the SOA paradigm. It consists of more than thirty systems, with independent lifecycles, which intercommunicate through HTTP APIs. Most of those APIs are fully compliant to the REST style. In cases when technology or business imposes restrictions to a representational approach, the REST style constraints may be loosened, but lower level platform and protocol guidelines are always kept as consistent as possible.



The Request Journey

The opening diagram present in this document depicts the journey of a request after it is sent by the user browser to a **VTEX** hosted web store. From architectural perspective, the path is the same for the web store itself and administrative services.

The main components depicted in the diagram are, as follows:

ELBs

Elastic Load Balancers are always the entry point to our services. Whenever a service is available through the Internet, it is actually the ELB that is publicly accessible. The servers themselves are not directly accessible.

Janus

Janus is our *API Manager*. It is an internally developed solution. It is consisted of a set of components and is responsible for aspects like HTTPS offloading, caching, request routing and some security concerns.

App Module

An *App Module* represents a functional component of the solution. It may be the Checkout Services or Logistics Service, for example. Actually, any of our services or worker processes may be represented the *App Modules* in this diagram. The only difference being that services have an Internet facing interface, GUI or API, and workers don't.

Data Services

The picture represents the main durable Data Services we use in **VTEX**:

- RDS (MySQL)
- SQL Server
- Amazon S3
- Dynamo DB

Also the messaging platform we mostly use: Amazon SQS.

Other services that might be considered data services are used as well, like Solr, Elasticsearch and Redis. But those are not our durable storage and are deployed more like the *App Modules*.

Security

VTEX did not have any security breaches, disclosures of Personally Identifiable Information (PII), alteration or damage of web server content or denial of service incidents as of the date of writing of this document.

Application Security

Authenticating Users in VTEX: VtexID

VTEX uses its own authentication system, called **VtexID**. The same system authenticates either Web Store users or Admin application users. The difference relies on the scope of the authentication, where the Web Store and the Admin application are represented by different scopes, each one with its users.

The usability of the Web Store, mostly driven by **SmartCheckout™**, doesn't stimulate the creation of a password. A user hardly needs to log in to the system. When they do, the first option presented is the creation of a one time token that is sent to the user's email. **VtexID** can also integrate to external authentication systems, as long as they implement *OAuth 2* protocol. Natively, **VtexID** offers *Google Account* and *Facebook* as external authentication systems. At last, but least used, the store may opt to provide the user with the ability to create a password.

The same user authentication options are available for the Admin application, but in a different scope from the Web Store.

Authenticating Applications

VtexID identifies not only users, but applications as well. In fact, applications run in the authentication context of a user. The difference is the kind of credentials used. When sending a request to one of VTEX's APIs, an application should use a pair of credentials consisting of an application key (**AppKey**) and an application token (**AppToken**). These credentials are issued for a user and they can use them to delegate their permissions to some application.

Authorization

Users, as well as applications, are authorized by VTEX based on access configuration done by an account administrator. This is done using a module called **VTEX License Manager**. This authorization is based on a User/Profile/Resource model.

Data Encryption: Key Strength & Algorithms

Passwords are stored as an SHA-256 hash.

Retrievable data like, for example, credit card information, are stored using RSA encryption with 2048-bit keys.

Periodic key rotation is performed for credit card information. The crypto period is not fixed and varies to a maximum of 6 months. The rotation is performed with no human intervention by an automated process that is responsible for the whole task: from the generation of the new key to the update of all cryptograms to the newly generated key.

Infrastructure Security

Firewalls Policies & Rules

Besides OS local firewalls on servers, we rely on AWS Security Groups, ELB configuration and VPC routing to obtain firewall functionality.

References:

- <http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/using-network-security.html>
- <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html>
- http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html
- <http://aws.amazon.com/documentation/elastic-load-balancing/>
- <http://aws.amazon.com/documentation/vpc/>

Since the deployment of our applications is done automatically and every log is collected and concentrated, even our staff hardly needs day-to-day access to our servers. The usual access to production environment occurs using AWS console or our own web applications and APIs. For that reason, our Security Groups and ELBs are configured so that only HTTP and HTTPS access is allowed for ELBs that expose some published feature. When some additional access is eventually necessary, it may be requested and temporarily provided for the specific purpose.

Application deployment and logging are subjects for other sections of this document.

All AWS API calls, which includes those coming from AWS Console, are logged using AWS CloudTrail (<http://aws.amazon.com/documentation/cloudtrail/>) for auditing.

Vulnerability Patch Management

There is more than one channel through which threats and security vulnerabilities may be acknowledged by **VTEX**. The most notable are **IDS** alerts, the periodic security scans, **AWS** reports, platform vendor reports, security scans run by some of our customers and the monitoring of renown entities as **CSIRT**, **CAIS** and **CIS**.

Security patches considered to be critical should be applied in no more than 30 days after the have been published by vendors of appliances and software managed by **VTEX** in **AWS**'s environment or on devices that have direct to access this environment. This responsibility is shared between **VTEX** and **AWS**. Both companies keep a constantly open channel in order to have security concerns promptly communicated and plans put in place for the definitive application of required patches or compensatory measures are provided while those are not yet available or the time needed for its application is considered too long and another temporary solution might be available in a shorter time.

Server OS Hardening

VTEX always creates its environments using the most recent **AMI** provided by **AWS** for each deployment service, be it *Elastic Beanstalk* or *OpsWorks*. By doing this we leverage our security on the hardening **AWS** already provides for instances deployed by their services.

We complement this hardening with the following measures:

- Applying critical security patches to the OS when they are not provided as an updated AMI by AWS;
- Centralizing system logs, so we don't lose log information when a virtual server is terminated for any reason;
- Monitor configuration files changes using Splunk to notify us about those changes;
- Have local firewalls configured with only HTTP(S) ports opened by default;
- Block traceroute;
- Block telnet;
- Block idle sessions after 15 minutes;
- Install antivirus software.

All these actions are automated during deployment and the artifacts that describe them are the deployment automation scripts themselves. In case a new module needs some additional software installed in its servers, both installation and hardening for that software must be added to the deployment automation scripts, in order to grant all actions are executed every time a new instance is created to host the given application.

Information Security Policy

VTEX faces the protection of its or its clients' information as a core concern. Information assets are considered strategic for corporation management and opportunities. This policy holds as its goals:

- A.** Formally express the directions for Information Security at VTEX as a mean to preserve:
 - a. Confidentiality:** information should be available only to those authorized to it;
 - b. Coherence:** information must be coherent, consistent and uncorrupted during the creation, processing and discarding cycle;
 - c. Availability:** information must be available whenever it is needed for the flow of business processes.
- B.** Guide the conversion of security management strategies into action.
- C.** Demonstrate the commitment of the company to the security of information.

Attachment: [“VTEX Information Security Policy”](#)

Development & Deployment

Each of **VTEX Commerce Suite**'s components is built as a service, or a worker process, or a composition of both, being a service an application component available on the web and a worker process a component responsible for dealing with event based, time based, or maintenance tasks. We will refer to one of those components as an **App Module**.

Team Governance

We keep a small team, from 2 to 4 members, fully responsible for one or a set of those *App Modules*. This team improves maintains and operates the modules under its responsibility. When a demand for a new version of the application is identified, the development tasks are created. After the code is written and tested by our QA people and automated processes, it is cleared for publication.

Each team is also empowered with the autonomy to choose the development approach that better suits the characteristics of the problem to be solved, comprehending development standards, platform, programming language or testing strategy. The only constraints enforced by the company are in the realm of keeping common standards for aspects related to the relationship between services, monitoring and governance.

Continuous Deployment

At the time of publication, a fully automated process is triggered and a new environment is created for the new version of the software.

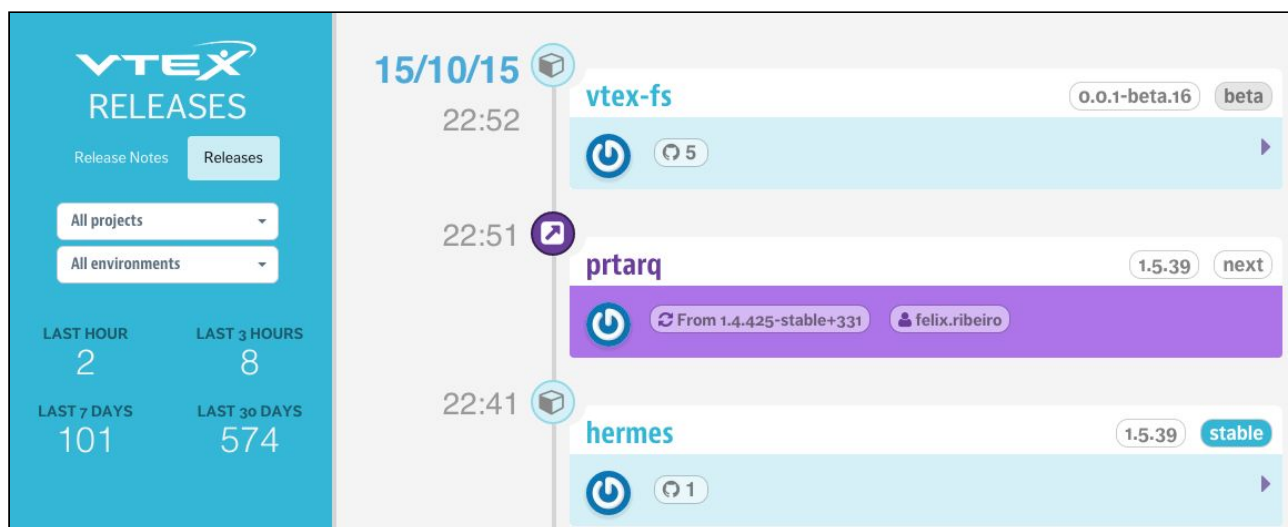
The environment creation process is implemented to guarantee it will have platform and OS versions and configurations compliant to the latest security and performance definitions held by the company.

When the environment is in place, the new software is installed and Janus, our API Manager / Router, is notified so it starts to send requests to the new Load Balancers.

Through monitoring, the module's team determines the new version is healthy and terminates the old environment. In case some issue is shortly detected on the new version

in production, the old environment is not terminated and Janus is notified about the version rollback, in order to turn the requests towards the previous version, allowing the team to safely assess the issue and work on the fix for a new publication round to start.

The following picture shows a glimpse of our releases monitor.



Monitoring & Testing

Unit Testing

Unit testing strategy and form of implementation is one of the aspects for which the decision is delegated to each of the development teams in **VTEX**. No common approach is enforced as one may not fit all solutions. Whenever an implementation for unit tests is spread across teams is due to the collaboration between them in showing what worked and what didn't work for each of them.

Integrated Tests

The common layer of tests is that of the integrated tests. They are performed from two points of view: the API consumer and the GUI consumer.

In the API consumer realm, our QA team implements tests that validate the proposed behaviors for the APIs and include them in the automated testing harness. Some of those tests are executed for every build of the service which it applies to and the whole set is executed nightly.

For the GUI consumer usage tests, the QA team automates service usage scenarios and they are constantly repeated by a robot that acts as a real user performing usual or critical actions on the system and verifies the results against expectations.

Acceptance Tests

When a new feature is under development by one of the teams, it is described to QA team and test planning begins. The feature is usually tested manually as a way of capturing all aspects of usage, of GUI or API, depending on the case. These manual tests work as a first layer of testing and, after that, those scenarios are automated and included in the test harness.

Regression Tests

Every test that was automated and included in the harness turns into a regression test and is now executed repeatedly for every build of the system, or nightly, depending on the criticality of the scenario.

Load Tests

Motivated by the single event with the biggest load for the e-commerce, the **Black Friday**, **VTEX** has put in place a framework for load testing the whole "customer journey" in the web store. Every year, eight weeks before the Black Friday, a calendar of tests is created and every development team in the company is involved. The goals for the tests are defined by a function of several variables:

- the actual numbers of the previous Black Friday;
- the numbers of this year, up to this date, compared to the same periods in the previous year;
- the increase in the number of customers hosted by **VTEX** and the impact those new customers had in the current numbers.

During this period of eight weeks, several rounds of tests are executed. For each round, the behavior of each *App Module* is assessed and adjustments are made to prepare for the next round. After the goal is reached, if there is enough time, more aggressive numbers are proposed and a new goal may be set as chance to find even more opportunities of optimization.

Besides the periodic load tests performed aiming at the Black Friday, any time a team responsible for some *App Module* may request a specific load test session to assess the code for some important change made in their solution. The methodology applied is always very similar to that developed for the Black Friday preparation, but the scope becomes more specific and focused.

Logging

VTEX collects application logs and system logs, aggregating them on Splunk (<http://www.splunk.com>).

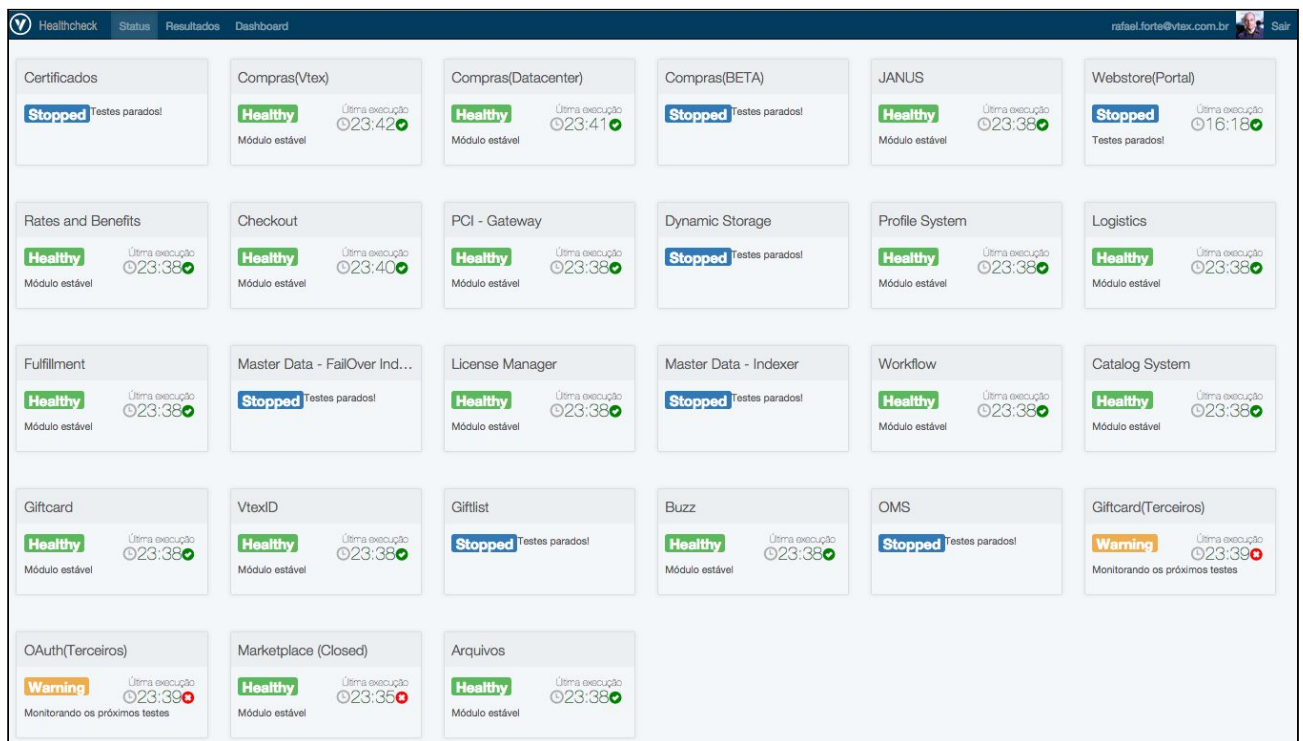
Application logs contain mostly metrics and errors, while system logs include OS application, system and security logs, antivirus logs. System logs collection is more comprehensive depending on the sensitivity of the data handled by each application.

For systems that handle more sensitive data, like credit card data, all logs are collected and alerts are configured on Splunk for certain kinds of entries. Also IDS tool has alerts configured. This way we have the team notified in the case of potentially relevant incidents.

The Health Check Robot

A key component of our monitoring system is the Health Check Robot. It tests, amongst other things, the consumer journey in the web store. It constantly repeats purchases in the system as if it were an end user of a web store. Three issues in sequence raise an alarm and a notification is dispatched to the whole company so the assessment of the incident starts.

Two kinds of incident are considered issues: when an error occurs during the test; when some feature takes more time than expected to complete. Delays in the execution are reported as warnings and three warnings raise an alarm just as three errors.



Healthcheck	Status	Resultados	Dashboard	rafael.forte@vtex.com.br	Sair
Certificados	Stopped	Testes parados!			
Compras(Vtex)	Healthy	Última execução 023:42 Módulo estável			
Compras(Datacenter)	Healthy	Última execução 023:41 Módulo estável			
Compras(BETA)	Stopped	Testes parados!			
JANUS	Healthy	Última execução 023:38 Módulo estável			
Webstore(Portal)	Stopped	Testes parados! Última execução 016:18			
Rates and Benefits	Healthy	Última execução 023:38 Módulo estável			
Checkout	Healthy	Última execução 023:40 Módulo estável			
PCI - Gateway	Healthy	Última execução 023:38 Módulo estável			
Dynamic Storage	Stopped	Testes parados!			
Profile System	Healthy	Última execução 023:38 Módulo estável			
Logistics	Healthy	Última execução 023:38 Módulo estável			
Fulfillment	Healthy	Última execução 023:38 Módulo estável			
Master Data - FailOver Ind...	Stopped	Testes parados!			
License Manager	Healthy	Última execução 023:38 Módulo estável			
Master Data - Indexer	Stopped	Testes parados!			
Workflow	Healthy	Última execução 023:38 Módulo estável			
Catalog System	Healthy	Última execução 023:38 Módulo estável			
Giftcard	Healthy	Última execução 023:38 Módulo estável			
VtexID	Healthy	Última execução 023:38 Módulo estável			
Giftlist	Stopped	Testes parados!			
Buzz	Healthy	Última execução 023:38 Módulo estável			
OMS	Stopped	Testes parados!			
Giftcard(Terceiros)	Warning	Última execução 023:38 Monitorando os próximos testes			
OAuth(Terceiros)	Warning	Última execução 023:39 Monitorando os próximos testes			
Marketplace (Closed)	Healthy	Última execução 023:35 Módulo estável			
Arquivos	Healthy	Última execução 023:38 Módulo estável			

Compras(Datacenter) - Ultimos Resultados

Filtros:   Erros **2** |  Alertas **4** |  Sucessos **294** | Total **300** | Date: 2015-10-15 

Página 1 de 3 / Total de Items 300

[Anterior](#)

1

[2](#)

[3](#)

[Próxima](#)

Hora	Status	Caso de Teste	Modulo
15/10/2015 12:15:35		(Browser: chrome) Multiloja - Cartão - Usuario Cadastrado	Compras(Datacenter)
15/10/2015 12:12:51		(Browser: chrome) Boleto - Usuario Novo(PJ) Sem CPF	Compras(Datacenter)
15/10/2015 12:08:42		(Browser: chrome) Boleto - Usuario Novo(PJ) Com CPF	Compras(Datacenter)
15/10/2015 12:04:34		(Browser: chrome) Boleto - Usuario Cadastrado	Compras(Datacenter)
15/10/2015 12:01:30		(Browser: chrome) Cartão - Usuario Cadastrado - Compra Parcelada	Compras(Datacenter)
15/10/2015 11:58:30		(Browser: chrome) Cartão - Usuario Novo (CPF)	Compras(Datacenter)
15/10/2015		(Browser: chrome) Cartão - Usuario Cadastrado	Compras(Datacenter)

Disaster Recovery

Application

The recovery of the application, in case of a catastrophic failure of the environment, is directly related to the automation of the deployment. All we have to do to restore any version of any component of our system is point our deployment service to a healthy availability zone and start the deployment process.

Data

Backup policies vary from one module to the other, when it comes to periodicity of backups and retention period. Whenever backup is not a service provided by **AWS** as a configurable feature of a data service, it is done by **VTEX** and stored in a durable storage, like S3 or Glacier. Backups are done in periods that vary from 6 hours to daily, and retention periods are usually of one week.

Process of Information Deletion

Once information stored by **VTEX** is not **VTEX**'s property, but its customer's property, the policies and responsibility for the information deletion is held by the company that owns it. After any information is actively deleted by a **VTEX**'s customer through the available APIs, it is immediately removed from our repositories but from data backups. After the backup retention period, the deleted information is definitively erased.

Credit card data that was tokenized and stored for later use may be automatically deleted after it expires. No active request is needed in this case, as this information is not owned by the store, but by the card holder and **VTEX** only holds its custody.

Integration

VTEX's integration strategy is fully based on its APIs. Usually **VTEX** acts passively as an integration endpoint to be consumed by whichever allowed service that needs access to data stored or processed by **VTEX**.

For some integrations that are related to the online user experience, most specifically to the order lifecycle, and need to be available at the time of purchase, **VTEX** assumes an active role in the integration process by defining an open protocol, based on **REST** APIs, and places requests to the proper resources, in configured endpoints, in the proper time. Some examples of this approach are the *WMS* and the *Gift Card Provider* protocols.

For more details: <http://help.vtex.com/developer-docs/>

Regulatory Compliance

PCI DSS

VTEX payment solution (**VTEX PCI Gateway**) is a PCI certified system, and has been since 2013. We are certified by Cipher, the same company that certified Cielo, one of the most prominent acquirers in Brazil.



The most recent version of our certification statement can be found here:
<http://secure.vtex.com/img/certificado-vtex-full.png>.

Implemented in a segregated infrastructure, with its own team who are the only people who have access to this infrastructure. To guarantee the compliance of the whole platform, the solution is designed so that **VTEX PCI Gateway** is the only component that touches payment data.

SSAE 16

VTEX by itself is not SSAE 16 audited. Nevertheless, our IaaS provider, **AWS**, is reportedly compliant to SSAE 16 and, given some requirements are fulfilled, their audited report may be provided by request.

In May 2017, we've had our first pre-assessment for a SOC 2 Type II report. Grant-Thornton is the third-party doing the work. The planned schedule leads us to having our first report on the second quarter of 2018. From there on, we will have annual reports covering January to December every year.

IFRS

On accounting standards and controls, **VTEX** is **CPC PME** audited. **CPC PME** is, in turn, compliant to **IASB's IFRS for SMES**.