

Virto Commerce Architectural Guidelines



What is Virto Commerce?

Virto Commerce is an API-driven, headless ecommerce platform, designed to build large and complex digital commerce solutions for B2B, B2C or B2B2C business, marketplaces and derived SaaS commerce platforms.

The architecture of the platform consists of a Microsoft based stack of technologies, an Agile development process, and DevOps to overcome technological, business and organizational challenges in competitive markets.



Principals

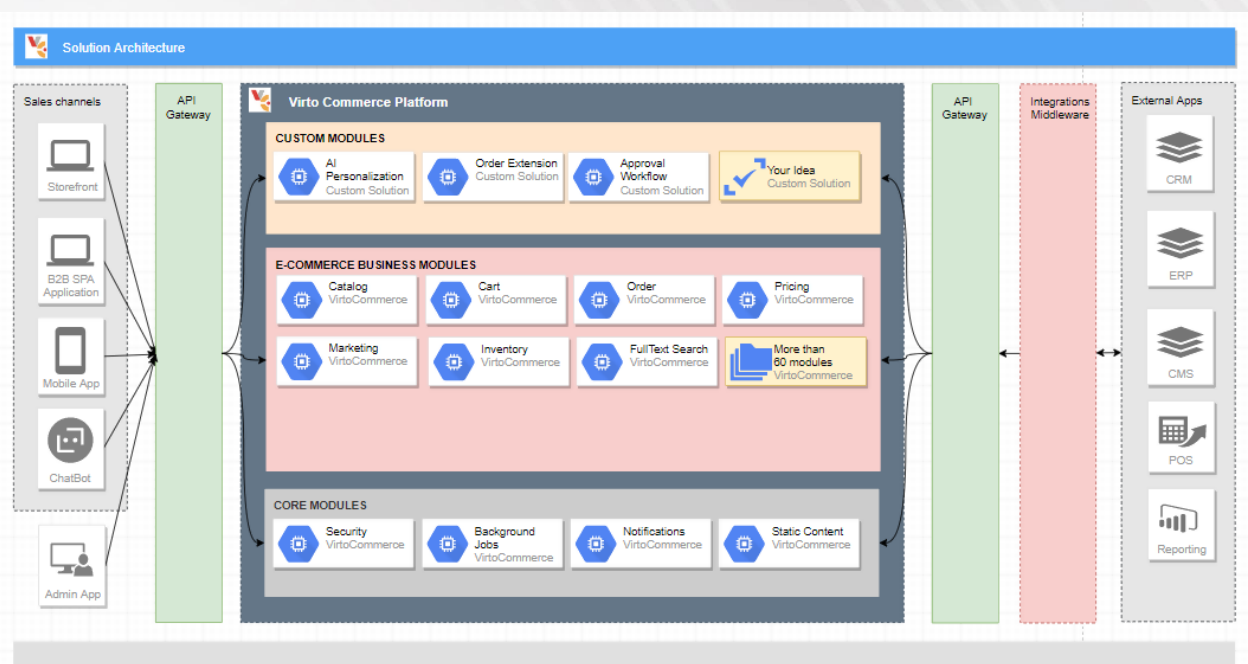
- **Enterprise-oriented** – ready for all types of ecommerce. No matter what size business.
- **Open source/transparency** - transparency of code and all development processes, an open product community hosted on GitHub.
- **Headless** - the content presentation layer is separated from the business logic and functional layer.
- **API-driven** – all methods are available via REST-API.
- **Modularity** – Virto Commerce has truly modular architecture. Such an approach allows getting all the benefits from an agile development process in a multi-dev teams environment. This makes feature development and the delivery process much faster, reduces the overall complexity of the system and as a result - lowers the cost of ownership.
- **Modern** – uses a modern stack of technology ASP.NET Core, EF Core, HTML 5, Elastic Search, Angular, Bootstrap 4.0, Progressive Web App (PWA).

- **Cloud-oriented** – originally designed and optimized to run in the MS Azure cloud, has seamless integration with native cloud services provided by the MS Azure ecosystem.
- **Scalable** – by design supports vertical and horizontal scalability.
- **Extendable** – custom solution development is accomplished without modification of the source code of the platform. There are several levels of extensibility such as replacement of existing modules, subsystems, or extension of existing functionality. Many subsystems such as catalogue management, order management, and others have a very flexible structure that allows implementing a broad variety of business requirements.
- **Evolutionary** - designed for evolution within depended feature development and frequent small releases - no monolithic solution with strongly developed CI/CD processes.
- **Seamless Integration** – is the process where a new module or feature of an application or hardware is added or integrated without resulting in any discernable errors or complications.



Architecture Overview

The following diagram illustrates the high-level architecture and main areas of Virto Commerce solutions.



These are the levels described in this diagram:

Sales channels (Content distribution) - represents the special touch points with user interfaces to access platform functionality for various groups of users.

Virto Commerce Platform - represents the scalable host of various installed modules and the main API facade for access to all functions and data that modules expose.

Integration Layer – seamless integration with 3rd party services via Azure Logic Apps or Dell Boomi.

Virto Commerce' s Platform is designed on Headless and API-driven architecture which describes the decoupling of the presentation layer of sales channels and ecommerce functionality.

Virto Commerce presents a lot of out-of-the-box functionality:

- **Core Modules** – to cover common functionality, like Security, Settings, Notifications, Assets etc.
- **Ecommerce Business Modules** – contains the implementation of Ecommerce business functionality which covers common business scenarios like Multi-stores, Multi-language catalogue, Multi-currency prices, cart, order, full-text search engine and more than 60 other modules.

The custom solution extends business functionality by adding custom modules:

- **Custom Business Modules** – 3rd party or custom modules which usually implement custom business functionality

The custom solution can add additional sales channels to search for and purchase products. Brands can achieve a more unified presence across devices and touchpoints such as mobile applications or POS terminals.



Technology stack

- ASP.NET Core as a base platform
- Entity Framework Core primary ORM
- ASP.NET Core Identity for authentication and authorization
- OpenIddict 2.0.0 for OAuth authorization
- WebPack as primary design/runtime bundler and minified
- Swashbuckle.AspNetCore.SwaggerGen for Swagger docs and UI
- SignalR Core for push notifications
- AngularJS as a primary framework for SPA
- HangFire for run background tasks



Platform extensibility and modularity

Maximizing extensibility has been our goal through all aspects of Virto Commerce development.

Virto Commerce offers several types of extensibility:

- Use module configuration to work with different scenarios.
- Build a new module to implement custom business functionality.
- Keep current API Contract and replace current functionality with your own implementation.

Sometimes, the configuration of the module can be a very complex task, and we recommend splitting the business logic and creating several simple modules, instead of a single big one.

Developers have complete access to the Virto Commerce source code to customize and extend all aspects to meet their needs. However, modifying the source code makes maintaining your customizations difficult as the application evolves over time.

Therefore, we recommend following the binary-based development process to configure customizations in projects as discrete modules and then expand your features by installing modules.

Each module can create following things when installing:

- New Web APIs methods
- Own UI views and elements in platform manager SPA
- New security permissions and settings
- Have own database tables and other infrastructure
- Extend or change existing functionality implemented in other modules

Some use cases of extensions which are supported by Virto Commerce:

- Extending the Product, Cart, Order or Contact type in existing with new fields without direct original code modification from your custom module.
- Extend already existing marketing promotions with new types
- Extend the existing payment, shipment or tax methods for integration with new payment gateway for your region
- Change the workflow of order processing with new actions or approvals



Headless

Virto Commerce platform is headless and API-based ecommerce. The content presentation layer is separated from the business logic and functional layer. All the functionality that the platform has is exposed as various Restful API' s that fully satisfy the main principles of SOA (Service-oriented architecture).

You can increase the number of channels to search for and purchase products while having a unified presentation across devices and touchpoints. With a headless commerce platform, you don' t have to worry about re-architecting the platform when adding and selling across new channels.

Primary benefits from using a headless approach:

- Faster development times and reduced costs.
- It is easy for front-end developers to make updates and customizations.
- Use both direct and indirect channels.
- Improve personalization of the content.



Sales Channels Kits

Developers have access to several Virto Commerce Storefront Kits which can decrease development and go-to-market time:

- **Storefront Web Application** – multi-store and multi-theme web-application based on Shopify theme.
- **Java-Script shopping cart** - represents catalogue and checkout which can be embedded on any web site or landing page.
- **Mobile Starter Kit** – to develop a mobile application
- **Chatbot Kit** – intelligent bot kit to interact naturally with your users on websites, apps, Cortana, Microsoft Teams, Skype, Slack, Facebook Messenger, and more.
- **Virto Commerce Admin** - represents the special touch points with user interfaces to access platform functionality for a various group of users.

Every Kit contains up-to-date best practices and solutions that we have deemed needed for the majority of projects we build.

The clients are not for a single type of platform, they can provide a shared experience with a variety of clients such as mobile applications, conversational clients, PWA (Progressive web application technology from google), and POS terminals.



Security

Virto Commerce was designed for maximum security. We think about security throughout the entire lifecycle of your application, from design and implementation to deployment and operations.

The platform has role-based security, permissions and an audit log. The data-storage and network protocols support data encryption on a separate level.

We use automatic tools for the code analysis to check security-related rules. The clear majority of security-related rules originate from established standards:

- CWE (<http://cwe.mitre.org/>)
- SANS Top 25 (<https://www.sans.org/top25-software-errors>)
- OWASP Top 10 (https://www.owasp.org/index.php/Top_10-2017_Top_10).

The Azure cloud hosting provides:

- Application gateway with web application firewall comes with protection from the latest OWASP vulnerabilities
- Penetration testing is performed internally in Microsoft and can be easily executed for a specific application
- Authentication and authorization in Azure App Service integrated with Azure active directory, Facebook, Google, Twitter, Microsoft accounts with multi-factor authentication, password policy enforcement, token-based authentication and auditing
- Layered Security Architecture is provided through an isolated environment deployed into Azure virtual networks with specific network security rules
- Role-based security to access the environment
- Encryption in transit - transport-level encryption, such as HTTPS when you transfer data
- Transparent data encryption (TDE) and column level encryption (CLE) for SQL databases
- Cross-Origin Resource Sharing (CORS) rules
- Transport Layer Security (TLS) protocol to protect data when it's travelling between the cloud services and customers
- Microsoft Anti-malware for Azure cloud services

Therefore, we recommend building security into your application, processes and organizational culture.



Performance and Scalability

The Virto Commerce platform and storefront kit can be easily scaled out.

The solution supports vertical and horizontal scaling, data consistency, peaks loading and load balancing. The solution can be hosted in multiple regions.

The implemented cache-aside pattern with using the distributed cache as a sync server for local caches on each instance allows it to keep the data for each actual instance.

Each platform module works with own infrastructure (DB, index, other resources) and shares nothing with other modules. This also allows building a flexible IT architecture for the solution.

The platform architecture allows deploying some set of modules independently (as microservices).



Maintainability and updates

The platform has modular architecture that supports receiving updates at runtime on a working system. You will be able to receive all new features and bug fixes without having to deal with code and IT infrastructure operations. In conjunction with robust CI/CD and QA processes on the VC team and each module has independent development processes and with frequent releases, you will receive desired updates often.

It is important for your solution to always be up to date with the latest platform and modules versions.



Availability and recoverability

Every architect fears having their architecture go down with no way to recover it.

The Virto Commerce platform is designed in a way that anticipates failure at all levels. Part of anticipating these failures is designing a system that can recover within the time required by your stakeholders and customers.



Seamless integration

In digital transformation, B2B and B2C ecommerce applications are only one element in the business ecosystem but they should work together as one solution.

Virto Commerce supports integration with external services and helps decrease development and go-to-market time. New services like Marketing, Analytics, Loyalty, Payments etc. can be added or integrated without resulting in any discernable errors or complications. Stakeholders can be assured that no problems will result from the integration.

Virto Commerce provides:

- Extension points for Payments systems, Shipment system, Tax providers and 3rd-party/external services.
- Open REST API for integration with Commercial services: CMS, Promotions, Loyalty management, Price management, Customers etc.
- Open REST API for integration with Operation services: Catalog, Account, Order, Inventory, Reports etc.
- Open REST API for integration for Legacy systems: CRM, ERP, CMS etc.

These scenarios are supported by Virto Commerce:

- Connect new payment, shipment methods
- Connect custom tax providers
- Sync ecommerce orders with an ERP system.
- Sync back the ERP Order statuses and Invoices with ecommerce.
- Import the Catalog master data from PIM to ecommerce.
- Prepare content in CMS and publish to ecommerce.
- Implement approval process for documents, invoices, budgets, and purchase orders.
- Launch two different ecommerce solutions (Legacy and Virto Commerce) together without crashing the business process.
- Incremental update and smooth migration from the Legacy system.

Quick peer-to-peer (P2P) integrations can turn into a large headache. When your infrastructure only has a few components. P2P integration can seem like a lightweight way to connect everything together.

Virto Commerce was built by using integration middleware.

The integration middleware helps reduce and manage:

- Security Concerns – the middleware is central governance for data security and compliance
- Budget Constraints – when delivering a new application or new system to an enterprise, it's easy to extend the workflow and new items into the pipeline. There are no development efforts and hidden integration costs.
- Unlimited Capabilities – the middleware is described in open JSON-format, no-hardcoding business rules or data-transformation. A lot of ready to use connections with databases, messaging services, data services, applications and network protocols.

- Supportability & Scalability - the middleware has an out of box asynchronous processing, versioning, retry policy, error handling, monitoring, alerts and logging.



Enterprise ecommerce challenges

Virto Commerce platform is designed to answer enterprise ecommerce challenges:

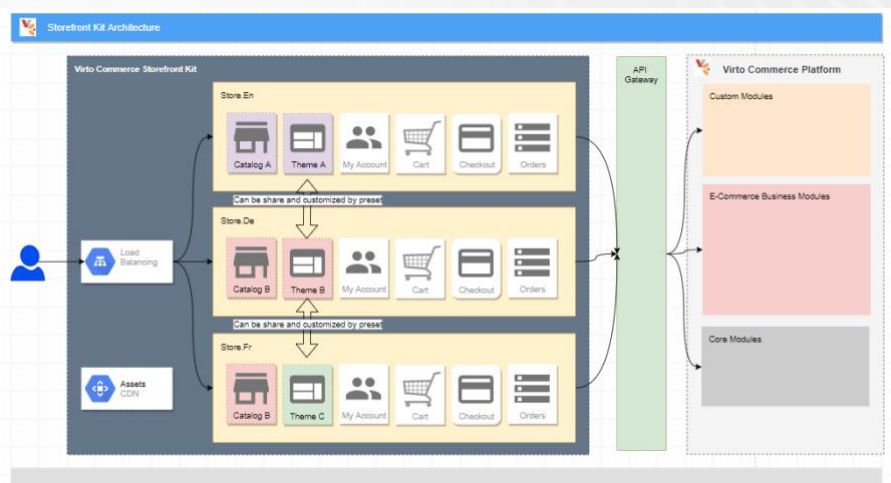
- Control performance key metrics (throughput, response time, max user load).
- How a data size is related to performance.
- How to support large catalogues (internal, external).
- Integration with different search engines (Elastic Search, Elastic Cloud, Azure Search, on-premise Lucene)
- How to grow the solution from Small, Medium to High load (enterprise).
- Authentication/Authorization (SSO)
- Modularity
- Control ecommerce KPI
- Scale solution by multiple geo regions, vertical and horizontal scalability, data consistency, peaks loading and load balancing.
- Health monitoring



Virto Commerce Storefront Kit

Developers can use the Virto Commerce Storefront Kit to create their own storefront application. The website is a client application for VC Platform and uses only public APIs while communicating.

The kit contains up-to-date best practices and solutions that we have deemed necessary for the majority of projects we build.



Key Features of the Storefront Kit:

- **Multi-Store support** – host multiple sites on the environment.
- **Multi-Language support**
- **Multi-Currency support**
- **Multi-Themes support with presents** – control layout, styles, features
- **Ready-to-use and Customize:** Home, Catalog, Cart, Checkout, MyAccount pages
- **Marketing and Dynamic Content support**
- **Social Login**
- **Faceted search support**
- **SEO friendly routing**
- **Mark-down pages support**

A theme is a main part of the Virto Commerce storefront application which provides a visual design for the entire storefront application area using a combination of custom templates, layouts, styles and images. Developers can customize out-of-box B2C and B2B themes.

Technologies and frameworks used:

- ASP.NET MVC Core
- ASP.NET Identity
- REST services clients generation with using [Microsoft AutoRest](#)
- Liquid view engine based on [DotLiquid](#)
- [LibSassHost](#) for processing **scss** stylesheets in runtime

The application has an extension model based on direct solution code changes. Developers can easily customize it by connecting with new modules via Rest API.



IT infrastructure and DevOps

When we talk about hosting, we are focused on cloud platforms. We have deep knowledge and experience using PaaS from Microsoft Azure. Our engineers take all the best from Azure to build fast, secure and modern solutions. Our platform has scalability options allowing us to build complex solutions from an infrastructure perspective with geo-replication and horizontal scaling to add resilience and redundancy to the whole solution.

We are an open source platform that you can easily deploy to Azure in a single click from the GitHub repository or/and go into details and review all the steps provided.

Our minimal recommended infrastructure is the following: 1 app service plan with 1 web app for backend and separate 1 app service plan with 1 web app for storefront applications, AzureSQL as a database, Elasticsearch as a search engine, Azure blob storage as a assets storage.

It is a starting point for all custom solutions. Depending on the client and their specific requirements, we may add necessary services. Here are the basic use cases:

- to increase security, we use an application gateway with a web application firewall and the number of security rules and policies
- to scale the solution, we use Redis cache, and web app autoscaling options
- to decrease the load on frontend we use CDN
- to decrease the load for backend we split logic between separate backend applications, where one group of backend apps are used by the storefront, and another one is used by jobs, integrations and other tasks

We are not limited to the mentioned use cases and are very flexible. By using different services, we can achieve client goals.

At the same time, we are not saying that Microsoft Azure is the only platform where you can implement your Virto Commerce solution. Amazon or Google Cloud can be easily used as well. Moreover, we are constantly researching and testing new technologies like Docker, Kubernetes and others to achieve better results and cover different scenarios. Currently, we are working on a method which will not limit us to use the Windows operating system, as we' ve almost completed the migration to .NET Core, so we can easily run Virto Commerce on Linux containers.

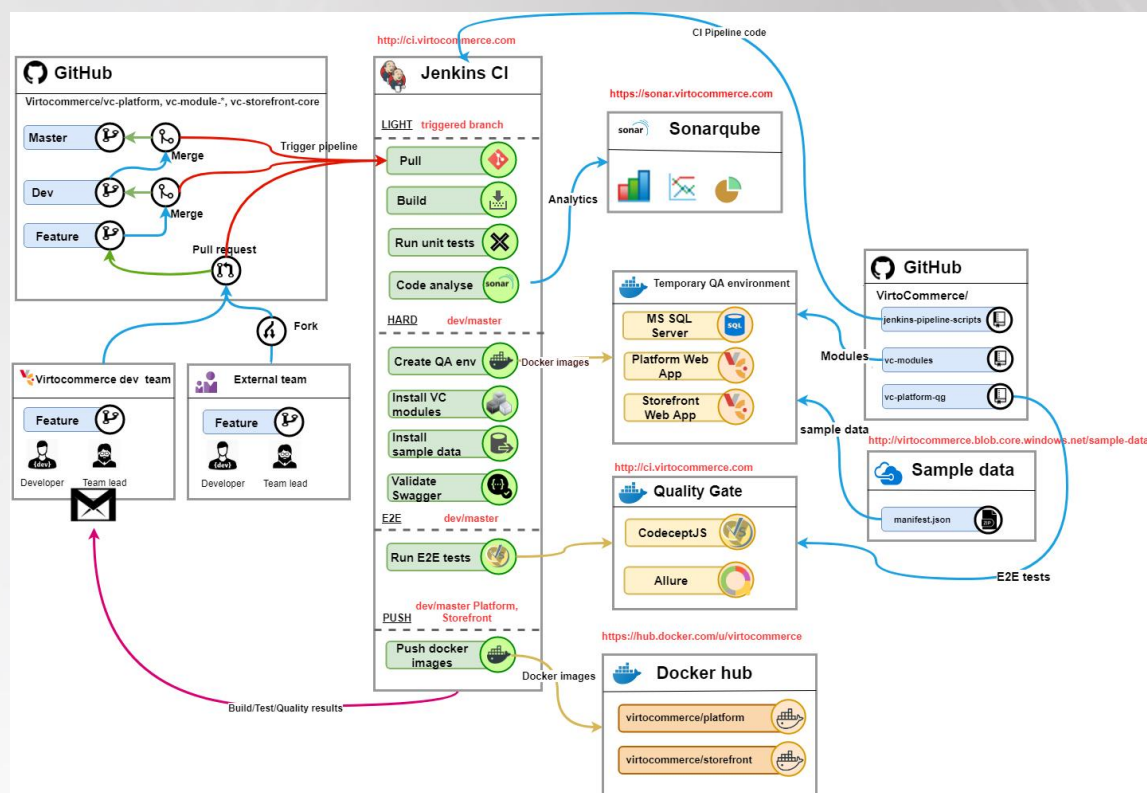
When we are talking about DevOps we follow the following principles:

- Mature processes with the ability to customize stages in a simple way
- Reliability of the solution through a transparent update process
- Git version control
- Jenkins as a main automation server
- All necessary scripts are available and publicly accessed in our GitHub repository
- Docker technology to easily and quickly create a temporary test environment
- Unit tests, integration test, performance test, end to end tests with reporting
- Infrastructure automation with infrastructure as a code

We apply those principles for a mature software lifecycle process. We are saying that to achieve better results we need 5 separate environments – they are:

- DEV – for developers, they have full control and ability to test their code in Azure
- QA – for QA team to run tests and check the quality
- QC – for BA to control the features that are implemented according to the requirements
- UAT – for User/Client acceptance
- PROD – live environment

The following diagram shows the current CI processes:





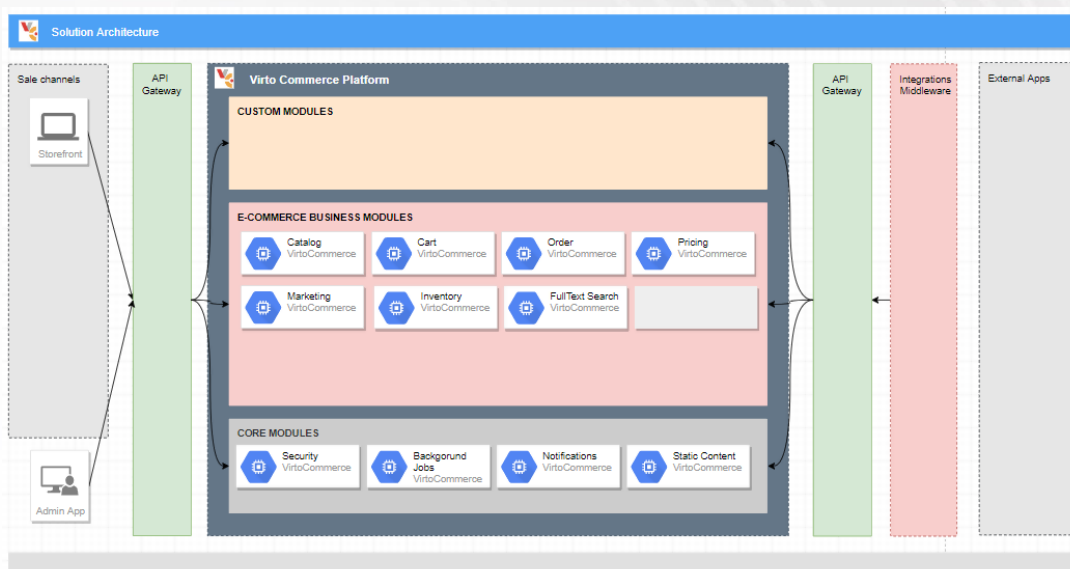
Custom Solution Development

When we talk about custom solutions we are focused on decreasing development and go-to-market time. These steps can help you to use our best practices.



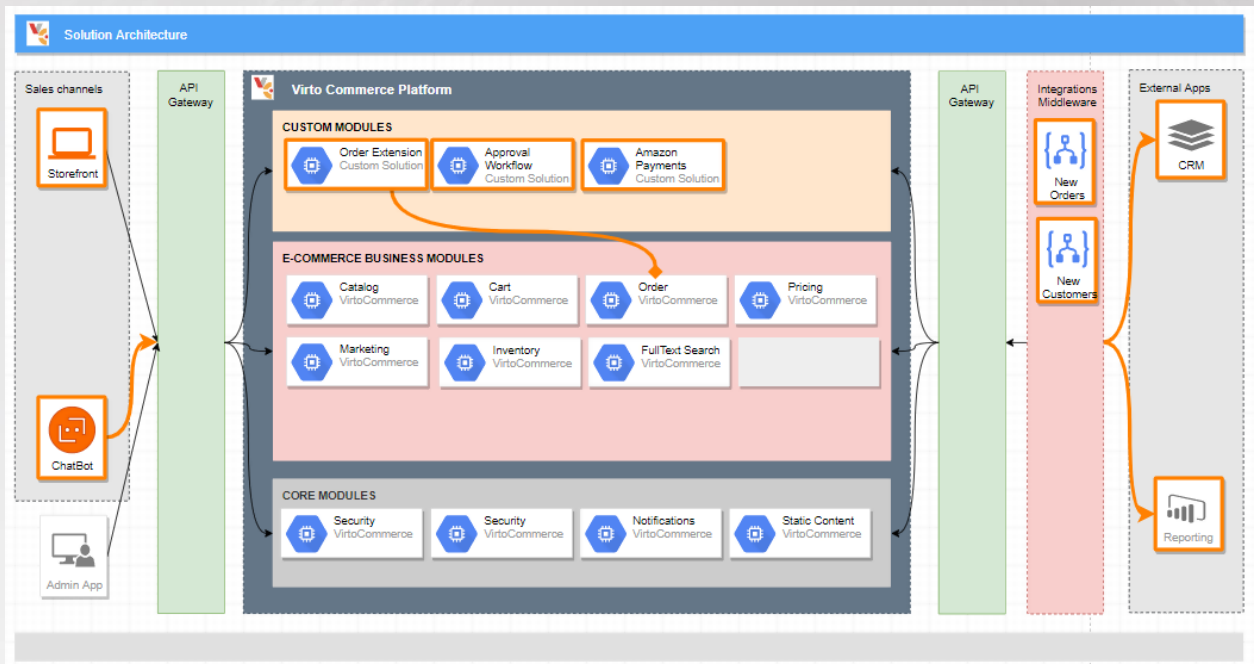
Step 1. Try to reuse as much as possible.

Virto Commerce provides out-of-box ecommerce business module and sales channel kits to save time and money. Every module has its own list of functional requirements to help business start to work instead of developing them from point-zero. The Design, Development, Quality Assurance and Release frameworks prevent fails and error in long term period.



Step 2. Customize the solution as you want.

Developers have complete access to the Virto Commerce modules to customize and extend all aspects to meet their needs. You need not build the whole the project. You only need to build your customizations. We recommend following the binary-based development process to configure customizations in the projects.



Step 3. Release and collect feedback faster.

Go to MVP and test your ideas without crashing the current business process.

Step 4. Scale up your solution with the business.

Virto Commerce is built for scaling and can grow up together with business.

Step 5. Keeping up to date.

Just like Web and the entire web ecosystem, Virto Commerce is continuously improving. Virto Commerce balances continuous improvement with a strong focus on stability and making updates easy. Keeping your Virto Commerce solution up-to-date enables you to take advantage of leading-edge new features, as well as optimizations and bug fixes.



Services Team

Most enterprise software requires some degree of implementation. Virto Commerce is no exception. Our professional services team provides implementation services in collaboration with partners. With any new, complex technology, expert services are essential to getting solutions up and running quickly and correctly, so customers can begin deriving value as soon as possible.